

Tarmo Kääriäinen

KVM-hallintatyökalut

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteet
Tradenomi, tietojenkäsittely
Kevät 2014

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Tarmo Kääriäinen	
Työn nimi KVM-Hallintatyökalut	
Vaihtoehtoiset ammattiopinnot Järjestelmänylläpito	Toimeksiantaja Timo Partanen
Aika Kevät 2014	Sivumäärä ja liitteet 77
<p>KVM (Kernel-based Virtual Machine) on yksi virtualisoinnin mielenkiintoisimmista hypervisoreista. Mielenkiinto sitä kohtaan kasvaa kun puhutaan kulujen hallinnasta. Ongelmaksi nousee kuitenkin sen hallinta, varsinkin kun kilpailijoina ovat VMwaren kaltaiset kokonaisvaltaiset ratkaisut. Tästä syystä useat yritykset ja yhteisöt kehittävät yhdessä avoimen lähdekoodin ratkaisuja KVM-hallintaan.</p> <p>Työn tarkoitus on selvittää, mitä ominaisuuksia ne pitävät sisällään, kuinka ne pärjäävät datacenterin hallinnassa, mihin käyttötarkoitukseen ne soveltuvat parhaiten, kuinka ne rakentuvat ja tarjoavatko ne käytännössä hyvän hallintaympäristön KVM:lle. Lähteinä on käytetty virallisia dokumentaatioita, wiki-sivustoja ja kirjallisuutta, joihin kehittäjät itse viittaavat ja tarjoavat.</p> <p>KVM-hallintatyökaluista testattavana ovat OpenNebula, oVirt, OpenNode, Proxmox ja Eucalyptus. Kyseiset hallintatyökalut helpottavat KVM-hypervisorin käyttöä datacenter- ja pilvi- ratkaisuihin. Kyseiset hallintatyökalut on valittu niiden teknologiaan ja käyttötarkoituksen erilaisuuden perusteella. Testit on suoritettu samanlaisella laitteistolla Kajaanin ammattikorkeakoulun tarjoamilla palvelimilla. Osa ominaisuuksista on testattu fyysisellä tietokoneraudalla ja loput vertailukelpoiset tulokset perustuvat VMwarella virtualisoituun ympäristöön, jonne hallintatyökalut on asennettu.</p> <p>Lopuksi päädytään tuloksiin, jotka kertovat testaajan mielipiteen perustuen testaajan käyttökokemukseen ja saatavilla oleviin dokumentaatioihin. Proxmox ja OpenNebula olivat testien onnistujia datakeskuksen hallinnassa, kun taas OpenNoden kehitystyö näyttää olevan kesken. Eucalyptus löysi paikkansa pilvestä ja oVirt haastaa Proxmoxin puhtaana virtualisointialustan hallintatyökaluna. Mahdollisia jatkotestejä olisivat käytännön testit, jossa ympäristöä käyttäisivät useat käyttäjät. Tällöin ominaisuudet ja käyttäjien hallinta nousisivat esiin käytännön tilanteessa.</p>	
Kieli	Suomi
Asiasanat	KVM, OpenNebula, OpenNode, Eucalyptus, oVirt, Proxmox, Virtualisointi
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Business Information Technology
Author(s) Tarmo Kääriäinen	
Title KVM Management Tools	
Optional Professional Studies System Administration	Commissioned by Timo Partanen
Date Spring 2014	Total Number of Pages and Appendices 77
<p>KVM (Kernel-based Virtual Machine) is one of the most interesting hypervisors in virtualization. Interest towards it grows when we talk about managing finances in virtualization. However, the KVM management might be troublesome. VMware has developed solid all-around software that is difficult to challenge. Still, there are many open source projects trying to facilitate the KVM management.</p> <p>The objective of this thesis is to answer a few questions about open source KVM management. What features do KVM management tools have? How well can datacenters be controlled using the current KVM management tools? How are these tools built up? What are the technologies? How are they meant to be used? Are they useful? Official documents, wiki-sites and books (recommended by developers) were used as sources.</p> <p>The management tools reviewed are OpenNebula, oVirt, OpenNode, Proxmox and Eucalyptus. They are meant to ease a datacenter and cloud management with the KVM hypervisor. These management tools are selected because they have different technologies and a different way of use. Still, they all have the same purpose. The management tools were reviewed in the Kajaani University of Applied Sciences. Some of the features were tested with physical servers, but the reviews are based on the results with VMware virtualized servers.</p> <p>The results are based on the opinion of the reviewer, but also on the technical documents of the developers. Proxmox and OpenNebula were the most promising tools for managing a datacenter. OpenNode was still in early development, so it did not do so well. Eucalyptus was clearly for the clouds and oVirt has a potential to challenge Proxmox as a non-cloud datacenter management tool. In future, there should be a practical test with a large number of users. That is the way to test the features and make a larger difference between KVM management tools.</p>	
Language of Thesis Finnish	
Keywords	KVM, OpenNebula, OpenNode, Eucalyptus, oVirt, Proxmox, Virtualization
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

SISÄLLYS

1 JOHDANTO	1
2 TEKNOLOGIAT	1
2.1 Virtualisointirajapinta	1
2.1.1 libvirt	1
2.1.2 Muut ratkaisut	1
2.2 VDSM	2
2.3 Fencing	2
2.4 Teknologiat virtuaalikoneiden käyttämiseksi	2
2.4.1 SPICE	2
2.4.2 VNC	3
2.4.3 SSH	3
3 OPENNEBULA 4.2	4
3.1 Rakenne	4
3.2 Asentaminen	5
3.2.1 Ubuntu 13.04	6
3.2.2 CentOS 6.4	7
3.2.3 OpenNebula Node	8
3.3 Käyttöliittymät	9
3.3.1 Komentorivi	10
3.3.2 Sunstone	10
3.4 OpenNebulan ominaisuudet	11
3.4.1 Klusteri	11
3.4.2 Hostit	12
3.4.3 Datastore	12
3.4.4 Verkot	13
3.4.5 Marketplace	13
3.4.6 Template	14
3.4.7 Levykuvat	15
3.4.8 OneFlow	16
3.4.9 Virtuaalinen datacenter	16
3.4.10 Käyttäjien hallinta	17

3.5 Virtuaalikoneen luominen	18
3.5.1 Levykuvan tuonti	18
3.5.2 Levyaseman luominen	18
3.5.3 Templaten luominen	19
3.5.4 Virtuaalikone	20
3.6 Teknologia	20
4 PROXMOX VE 3.1	22
4.1 Rakenne	22
4.2 Proxmox VE:n asennus	22
4.3 Klusteri	23
4.4 Subscription	23
4.5 Proxmoxin ominaisuudet	24
4.5.1 Käyttöliittymä	24
4.5.2 Migraatio	24
4.5.3 Käyttäjien hallinta	25
4.5.4 Autentikointipalvelin	26
4.6 High Availability	26
4.6.1 Two-Node Availability	26
4.6.2 Fencing	27
4.7 Datastore	29
4.8 Virtuaalikoneen luominen	30
5 OVIRT 3.3	32
5.1 Rakenne	32
5.1.1 oVirt Engine	33
5.1.2 oVirt Node	33
5.2 Asentaminen	33
5.2.1 oVirt Enginen asennus	33
5.2.2 Noden asennus	34
5.2.3 Valmis jakelu	34
5.3 oVirt noden lisääminen	35
5.4 oVirtin työkalut	37
5.5 Administration Portal	38
5.5.1 Guide Me	38
5.5.2 Käyttäjät	40

5.5.3 External Providers	40
5.6 User Portal	40
5.6.1 Virtuaalikoneen luominen	41
5.6.2 Template	41
6 OPENNODE 6	42
6.1 Rakenne	42
6.2 OpenNode OS 6:n asennus	43
6.3 Beta-asteella	45
6.4 Ominaisuudet	45
6.4.1 Template	45
6.4.2 Migraatio	46
6.4.3 Virtuaalikone	46
6.4.4 Storage	47
6.4.5 Web-UI	47
6.5 High Availability	47
6.6 Virtuaalikoneen luominen	48
6.6.1 Templaten luominen	49
6.6.2 Valmis virtuaalikone	49
6.7 Roadmap	49
7 EUCALYPTUS 3.4.2	51
7.1 Rakenne	51
7.1.1 Cloud Controller	52
7.1.2 Cluster Controller	52
7.1.3 Node Controller	53
7.1.4 Storage Controller	53
7.1.5 VMware Broker	53
7.1.6 Walrus	54
7.2 Asentaminen	54
7.2.1 Mukautettu asennus CentOSille	54
7.2.2 FastStart	55
7.2.3 Eucalyptus node	57
7.3 Asennuksen testaaminen	57
7.4 CloudWatch	57
7.5 Ylläpito-paneeli	58

7.6 Käyttäjä-paneeli	59
7.6.1 Virtuaalikoneen luominen	59
7.6.2 Virtuaalikoneen käyttäminen	60
7.7 Roadmap	60
8 POHDINTA	61
8.1 Hybridi-cloud	61
8.1.1 Eucalyptus	62
8.1.2 OpenNebula	62
8.1.3 OpenNode	63
8.2 Datacenter virtualisointi	63
8.2.1 Proxmox	63
8.2.2 oVirt	64
8.3 Käytettävyys	64
8.3.1 Eucalyptus	64
8.3.2 OpenNebula	65
8.3.3 OpenNode	66
8.3.4 Proxmox	67
8.3.5 oVirt	67
8.4 Asiakkaiden hallinta	68
8.5 KVM vastaan VMware	69
8.5.1 VMwaren vahvuudet	69
8.5.2 VMwaren heikkoudet	69
8.5.3 Avoimen lähdekoodin ohjelmistojen vahvuudet	70
8.5.4 Avoimen lähdekoodin ohjelmistojen heikkoudet	71
8.6 Testaajan tulokset	71
LÄHTEET	74

SYMBOLILUETTELO

Active Directory	Windows-toimialueen käyttäjätietokanta
Auto-scaling	Ominaisuus, joka säättää resursseja
Bare-metal	Suoraan tietokoneen rautaan yhteydessä oleva ympäristö
BIOS	Basic Input/Output System, tietokoneohjelma
Bugi	Virhe ohjelmistossa
CD	Compact Disc, tallennusmedia
CLI	Command Line Interface, komentorivi
EPEL	Extra Packages for Enterprise Linux, repositorio
Fencing	Tapa, jolla turvataan ympäristön toimivuus uhan toteutuessa
Front-end	Palvelin, jolla hallitaan nodeja
FS	File System, tiedostojärjestelmä
GID	Group Identifier, Ryhmän tunniste
HA	High Availability, korkea saatavuus
Host	Palvelin, joka tarjoaa palveluita
HPC	High-Performance Computing, supertietokone
Hybrid	yhdistelmä
ID	Identifier, tunniste
iDRAC	Integrated Dell Remote Access Controller, hallintatyökalu
IP	Internet Protocol, protokolla
IPMI	Intelligent Platform Management Interface, käyttöliittymä
iSCSI	Internet Small Computer System Interface, protokolla
KVM	Kernel-based Virtual Machine, virtualisointiohjelmisto
LDAP	Lightweight Directory Access Protocol, protokolla
Live	Reaaliaikainen tapahtuma

Load-balancing	Ominaisuus, joka lieventää kuormitusta
LVM	Logical Volume Manager, looginen taltiohallinta
Migratointi	Siirto-operaatio
NC	Node Controller, Noden hallitsija
NFS	Network Filesystem, protokolla
NIC	Network Interface Controller, verkkokortti
Node	Palvelin, joka liitetään toisella palvelimella hallittuun ympäristöön
OMC	OpenNode Management Console, hallintakonsoli
OMS	OpenNode Management Server, hallitseva palvelin
OS	Operating System, käyttöjärjestelmä
Patch	Korjaus ohjelmistoon
Portal	Portaali
RDP	Remote Desktop Protocol, protokolla
Roadmap	Suunnitelma
Snapshot	Virtuaalikoneen varmuuskopio
SPICE	Simple Protocol for Independent Computing Environments, protokolla
SSH	Secure Shell, protokolla
Storage	Tietovarasto
Subscription	Tilaus
Template	Muotti, josta voi tehdä useita samanlaisia virtuaalikoneita
Thin Provisioning	Ominaisuus, joka säästää levytilaa
TUI	Text User Interface, puoligraafinen käyttöliittymä
UEFI	Unified Extensible Firmware Interface, ohjelmisto
UI	User Interface, käyttöliittymä
UID	User Identifier, käyttäjän tunniste

VDSM	Virtual Desktop And Server Management, taustaprosessi
VE	Virtualization Environment, virtualisointiympäristö
VM	Virtual Machine, virtuaalikone
VMFS	Virtual Machine File System, tiedostojärjestelmä
VNC	Virtual Network Computing, protokolla

1 JOHDANTO

Virtualisointi on kehittynyt viime vuosina paljon, ja siitä osoituksena myös ilmaisten hallintatyökalujen määrä on suuri. KVM on ilmainen hypervisor, jonka hallinta ei ole käytännöllistä ilman kunnollista hallintatyökalua. Tarkoitukseni onkin selvittää, kuinka hyvin KVM ympäristön hallinta sujuu tämän hetken avoimen lähdekoodin työkaluilla ja olisivatko ne virtualisointiympäristön hallintaan käyttökelpoisia. Lisäksi otan kantaa siihen, ovatko KVM-hallintatyökalut VMwaren haastajia tällä hetkellä. Mikäli ylläpitäjä miettii sitä, mikä olisi vaihtoehtona avoimen lähdekoodin KVM-ympäristöön, niin tämä opinnäytetyö on siihen vastaus.

Testini on vertailupainotteinen, joten hallintatyökalujen valintaprosessissa pyrin ottamaan erilaisia vaihtoehtoja testiin mukaan. Ensimmäinen valintaperuste oli teknologia, johon KVM-virtualisointi perustuu. Suurin osa hallintatyökaluista perustuu libvirt-teknologiaan, joten jouduin valitsemaan bare-metal ratkaisuja sen lisäksi mukaan, koska muita virtualisointirajapintoja ei ole tarjolla. Toinen seikka valintaprosessissa oli ideologia, johon koko ohjelmisto perustuu. Osa KVM-hallintatyökaluista on virtualisointialustan hallintaan, kun taas toinen osa suuntaa pilveen. Ydinasiana on se, että testattavat eroavat toisistaan mahdollisimman paljon. Valintaprosessin jälkeen hallintaohjelmistoista testattavana ovat OpenNebula, OpenNode, oVirt, Proxmox VE ja Eucalyptus. Kuvassa 1 esitetään hallintatyökalujen ominaisuudet.

	Proxmox	OpenNode	oVirt	Eucalyptus	OpenNebula
Virtualisointi API	bare-metal	bare-metal	libvirt	libvirt	libvirt
Hypervisor	KVM OpenVZ	KVM OpenVZ	KVM	KVM Xen ESXi	KVM Xen ESXi OpenVZ
Avoin lähdekoodi	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
High Availability	Kyllä	(Kyllä)*	Kyllä	Kyllä	Kyllä
Live Migraatio	Kyllä	(Kyllä)*	Kyllä	Kyllä	Kyllä
Storage	LVM iSCSI NFS GlusterFS RBD/Ceph Sheepdog	LVM	NFS iSCSI Fibre Channel LVM GlusterFS	iSCSI LVM NFS Fibre Channel	Fibre Channel NFS LVM Ceph iSCSI
Amazon Web Services	Ei	Ei	Ei	Kyllä	Kyllä
Load Balancing	Ei	Ei	Kyllä	Kyllä	Kyllä
Auto-Scaling	Ei	Ei	Ei	Kyllä	Kyllä
Verkon virtualisointi	Open vSwitch	(Open vSwitch)**	Open vSwitch	Open vSwitch	Open vSwitch
Käyttäjien autentikointi	Proxmox VE LDAP Linux Active Directory	LDAP Linux	LDAP Linux Active Directory	LDAP Linux Active Directory	LDAP Linux Active Directory
Julkaisu v1.0	2008	2010	2012	2008	2008
	*Merkillä varustettu ominaisuus on puutteellinen	**Merkillä varustettu ominaisuus on tulossa lähiaikoina			

Kuva 1. Vertailutaulukko testiin osallistuvista työkaluista.

Tutkimuksessani huomioin asennusprosessin etenemisen monella distribuutiolla ja asennustavalla. Asennusprosessi vaikuttaa yhdessä konfiguroinnin kanssa testieni lopulliseen tulokseen ja "testaajan valintaan". Erilaisten asennustapojen kautta tuon esille mahdollisia ongelmatilanteita esimerkiksi toista distribuutiota käytettäessä.

Käyttöliittymä on tärkein osa hallintatyökalua, koska se on suurin syy hallintatyökalun käyttämiseksi. Siksi perehdyn mahdollisiin ongelmiin käyttöliittymässä ja sen hallinnassa. Kaikki käyttöliittymät ovat pääosin graafiselta puolelta testattuja, sillä se on käyttäjäystävällisempää.

Ominaisuudet ovat tärkeässä osassa silloin, kun valitaan ympäristöön sopivaa hallintatyökalua. Välttämättä kaikki ominaisuudet eivät ole saatavilla kaikissa työkaluissa. Osa työkalujen kehittäjistä vielä kehittää ominaisuuksia, jotka ovat toisella olleet jo vuosia. Niillä on suuri vaikutus lopputulokseen ja siihen, millaiseen ympäristöön ne soveltuvat.

Ideologia virtualisointialustan hallitsemisessa on myös suuri asia valintaprosessissa. Otan kantaa siihen, millaiseen ympäristöön mikäkin työkalu soveltuu parhaiten. Vaikka kaikki KVM-hallintatyökalut olisivat samalla teknologialla toteutettuja, niin niiden käyttötapa saattaa olla erilainen.

Pohdintaosiossa otan kantaa siihen, mitkä ovat KVM-hallintaohjelmistojen vahvuudet ja heikkoudet verrattuna VMwaren tuotteisiin. VMware on yksi markkinoiden suurimmista virtualisointiympäristön hallintaan tarkoitetuista ohjelmistokehittäjistä.

Ympäristönä toimivat Kajaanin ammattikorkeakoulun käyttöjärjestelmälaboratorion palvelimet. Fyysiset palvelimet, joita käytän HA-ominaisuuden testaamiseen Proxmoxilla, ovat Dellin R410-sarjaa. Niissä on 16 GB keskusmuistia ja virtualisointia tukeva Xeon-prosessori. Muissa tapauksissa alustana on VMwaren virtualisointiympäristö, josta 4 gigatavua keskusmuistia on lohkaistu jokaiselle front-endille ja nodelle. Datastorena käytössä ovat fyysinen EMC CX3 iSCSI-storage ja Ubuntulle asennettu NFS-jako.

2 TEKNOLOGIAT

Pyrin käyttämään testeissäni erilaisia teknologioita käyttäviä hallintatyökaluja, jotta eroavaisuuksia tulisi enemmän esille.

2.1 Virtualisointirajapinta

Virtualisointirajapintoja hallintaohjelmistoja tutkiessa tuli vastaan ainoastaan libvirt. Toinen vaihtoehto sille oli bare-metal ratkaisut joissa rajapinta on toteutettu toisella tavalla. Rajapinnat mahdollistavat KVM:n käyttämisen yksinkertaisemmin. Käytännössä se mahdollistaa kaikkien hypervisoreiden käyttämisen samoilla komennoilla, eikä silloin tarvita hypervisorin omien komentojen opettelemista. (libvirt a 2014)

2.1.1 libvirt

libvirt on virtualisointirajapinta, jota useimmat hallintatyökalut hyödyntävät. Se tukee KVM- ja Xen-hypervisorin, joten sitä käyttävät hallintatyökalut tukevat myös samoja hypervisoreita. (libvirt b 2014)

2.1.2 Muut ratkaisut

Proxmox ja OpenNode käyttävät virtualisointirajapintanaan omia ratkaisujaan. Käytännössä kaikki KVM:n käyttämiseen liittyvät toiminnot ja ominaisuudet ovat tällöin kiinni kehittäjästä itsestään.

2.2 VDSM

VDSM on noden hallintaan tarkoitettu rajapinta. Se vastaa virtuaalikoneen elinkaaresta, hostin rekisteröimisestä, monitoroinnista, verkon konfiguroimisesta, storagen hallinnasta ja menettelytavoista liittyen virtualisointiympäristöön. (Kenigsberg 2011, 3-4.)

2.3 Fencing

Fencing on pakollinen toiminto, jotta HA on edes mahdollista toteuttaa. Käytännössä Fencing on mekanismi, jolla varmistetaan noden todella olevan "kuollut". Mikäli node onkin toiminnassa, niin nodelle lähetetään uudelleenkäynnistys tai sammutus signaali, jolloin ympäristöön ei synny useita instansseja samoista virtuaalikoneista. (Proxmox 2014)

R410-palvelimissa on IPMI-ominaisuus, jolla fencing on mahdollista ottaa käyttöön, joten HA-ominaisuuksien testaaminen on niillä mahdollista. Ominaisuuden saa päälle UEFI:n tai BIOSin kautta.

2.4 Teknologiat virtuaalikoneiden käyttämiseksi

Useimmat hallintatyökalut tukevat SSH-yhteyttä, mutta myös muita mahdollisuuksia löytyy.

2.4.1 SPICE

SPICE on Red Hatin kehittämä avoimen lähdekoodin työpöytävirtualisointiteknologia. Se tarjoaa korkealaatuisen työpöytäyhteyden QEMU-virtuaalikoneisiin. Useimmat projektissa testaamani hallintatyökalut tukevat kyseistä teknologiaa. (SPICE 2014)

SPICE toimii loistavasti KVM:n pohjalta tehtyyn työpöytävirtualisointiin. Käytännössä käyttäjä tuskin huomaa, jos virtuaalikone on täydessä ruudussa, että kyseessä on virtuaalikone.

2.4.2 VNC

VNC on protokolla, joka mahdollistaa työpöytävirtualisoinnin. VNC:n etu on Windows käyttöjärjestelmien tuki. Lisäksi suurin osa testeissä olleista hallintatyökaluista tukee tätä teknologiaa.

2.4.3 SSH

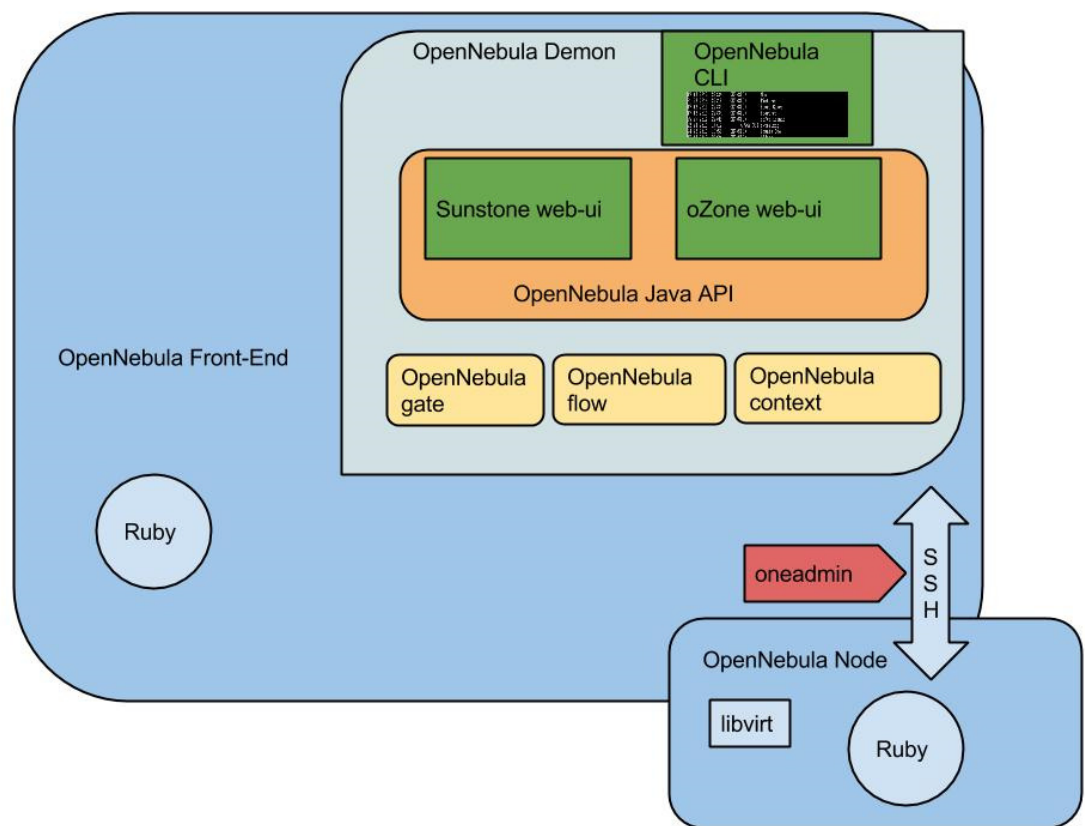
SSH eli Secure Shell on protokolla, jolla hallitaan Unix-järjestelmiä etänä komentoriviltä. Yleensä SSH-yhteys avataan kolmannen osapuolen ohjelmistoa käyttäen tai Unix-järjestelmän komentorivin kautta.

3 OPENNEBULA 4.2

OpenNebula on vuonna 2005 perustettu avoimen lähdekoodin projekti jonka perustajina toimivat Ignacio M. Llorante ja Ruben S. Montero. Ensimmäinen versio OpenNebulasta ilmestyi vuonna 2008. OpenNebulan tavoite on luoda avoin, joustava, laajentuva ja kattava hallintaympäristö, joka on suunnattu yritysten pilvipalveluksi. (OpenNebula a 2014)

3.1 Rakenne

OpenNebulan infrastruktuuri rakentuu front-endistä, hosteista, datastoreista ja virtuaaliverkoista (kuva 2). OpenNebulan palvelut toimivat front-endissä, josta se yhdistää salasananottomalla SSH-yhteydellä hosteihin. Jakelupakettien nimet ja sisällöt vaihtelevat versiosta riippuen, joten niiden kanssa tulee olla tarkkana.



Kuva 2. OpenNebulan paketit front-endin ja noden sisällä.

OpenNebulan taustapalvelut asentuvat opennebula-server paketilla. Kyseiset palvelut asentuvat ”/usr/bin/”-kansioon. Paketti nimeltään opennebula-sunstone sisältää web-käyttöliittymän front-endille. Virtuaalikoneiden ja OpenNebulan välisen kommunikoinnin mahdollistaa opennebula-gate-paketti. Palveluiden manageroinnista vastaavat opennebula-flow-paketin mukana tulevat palvelut. OpenNebulan toiminnan kannalta tärkeät ruby ja java tulevat paketeissa opennebula-ruby ja opennebula-java. Noden kannalta tarpeelliset työkalut ovat opennebula-node-kvm-paketissa.

Käytännössä OpenNebulalla on yksi palvelin, joka hoitaa kaiken manageroinnin. Palvelut tulevat useassa paketissa, mutta ovat riippuvaisia toisistaan siinä määrin, että kaikki on asennettava kerralla. Tämän johdosta front-end on yksi palvelin.

3.2 Asentaminen

OpenNebula ei vaadi kovin suuria määriä resursseja asentuakseen. Tryout-versiota on mahdollista testata jopa kannettavassa tietokoneessa järjestelmävaatimuksien puolesta. Palvelinversio vaatii OpenNebulan front-endille 3 gigatavua tallennustilaa ja ruby 1.8.7-version asennuksen. Hosti vaatii myös ruby paketin asennuksen sekä SSH-palvelimen. Virtuaalikoneiden luominen vaatii myös levyjärjestelmän tai palvelimelta omaa levytilaa. (OpenNebula b 2013)

OpenNebulan asentamiselle löytyy useita ohjeistuksia distribuutioiden wiki-sivuilta ja muista epävirallisista lähteistä. Käytin asentamisessa virallista ohjeistusta osoitteesta "<http://archives.opennebula.org/documentation:archives:rel4.2:ignc>". Tulevien uudempien versioiden asennusohjeet ovat siirtyneet osoitteeseen: "<http://docs.opennebula.org/>".

Aluksi asennan KVM1-palvelimelle ruby-paketin ja varmistamalla, että SSH on toiminnassa. Putty-ohjelmalla pääsen KVM1-palvelimeen ja etsin ruby paketteja komennolla “`aptitude search ruby`”. Ubuntun repositoriosta löytyy pitkä lista ruby-nimisiä paketteja ja valitsen niistä versionumeroltaan suurimman, eli “`ruby1.9.3`”. SSH-palvelin on valittu asennettavaksi jo Ubuntun asennusvaiheessa, joten sille ei tarvitse tehdä mitään vielä.

3.2.1 Ubuntu 13.04

Asennan OpenNebula 4.2:n Ubuntun 13.04-palvelinversioon. Kyseinen asennus tulee olemaan Front-End, jossa OpenNebulan tärkeimmät komponentit tulevat olemaan. Vaatimusten mukaisesti asennan rubyn myös front-endille.

Ensimmäinen askel on json-gemin asennus. “sudo gem install json” -komennolla saan kuitenkin virheen, jossa kerrotaan, että makefileä tehdessä kohdattiin virhe. Virhe on minulle ennestään jo tuttu muista Linuxiin liittyvistä asioista, joten se tarkoittaa sitä, että build-essentials-paketti on asentamatta. “Sudo apt-get install build-essential” -komennolla saan asennettua puuttuvan palikan ja jsonin asennus lähtee jatkumaan uudella yrittämällä.

Seuraavassa kohdassa asennusohjetta kehoitetaan uudelleen nimeämään version 1.8 json.rb-tiedosto nimelle json.rb.no, mutta näyttää siltä, että 1.8 versiota ei ole olemassa, vaan 1.9.1-versio on ainoa. Jätän tämän vaiheen tekemättä, ellei siitä koidu ongelmaa.

Lataan ja lisään OpenNebulan tarjoaman repo.key paketin, jotta voin lisätä ja päivittää OpenNebulan repositorion palvelimen sources.list.d-listaan. Sen jälkeen noudan ja asennan seuraavat OpenNebulan komponentit:

opennebula-common

opennebula-node

opennebula-sunstone

opennebula-tools

opennebula-gate

opennebula-flow

opennebula

Jätän asentamatta ohjeessa olevan libopennebula-ruby-komponentin, sillä sitä ei enää repositoriossa ole. Aptitude search -komento paljasti sieltä löytyvän ruby-opennebula ja libopennebula-java-komponentit, jotka olivat jo asentuneet.

Seuraavaksi suoritan `install_gems`-skriptin, joka asentaa tarvittavia kirjastoja OpenNebulalle. Komento on “`Sudo ./usr/shares/one/install_gems`”.

Asennus onnistui, sillä OpenNebulan palvelu on tavoitettavissa selaimen kautta osoitteesta `http://<Front-End-IP>:9869`.

3.2.2 CentOS 6.4

Seuraavaksi asennan CentOS 6.4-version virtuaalikoneeseen, josta on tarkoitus tulla front-end. Virtuaalikone toimii KVM3-palvelimella, jota käytän ympäristön rakentamiseen.

Myös CentOS tarvitsee ruby-paketin, joten suoritan komennon “`yum search ruby`”. Listasta löytyvä paketti “`ruby.x86_64`” vaikuttaa oikealta, sekä asennus kertoo asentavansa ruby version 1.8.7.

Etenen asennusohjeen mukaisesti lisäämällä repositorion EPEL, jossa OpenNebula 4.2 sijaitsee ja asennan seuraavat paketit `yum`-komennolla:

`opennebula-server`

`opennebula-sunstone`

`opennebula-ozones`

`opennebula-ruby`

`opennebula-java`

`opennebula-gate`

`opennebula-flow`

`opennebula-node-kvm`

Asennuksessa ilmenee ongelmatilanne, kun järjestelmä ei löydä seuraavia paketteja:

rubygem-sequel

rubygem-sqlite3-ruby

rubygem-json

rubygem-nokogiri

rubygem-sinatra

rubygem-rack

rubygem-uuidtools

rubygem-thin

Korjatakseni tilanteen haen “yum search rubygems”-komennolla uusia paketteja. Haku löytää paketin nimeltä “rubygems.noarch”, jonka myös asennan. Kyseinen paketti mahdollistaa gem-pakettien asentamisen.

Rubygems.org-sivustolta on saatavissa asennusohjeet tarvittaviin gem tiedostoihin, joten jokaisen kohdalla asentaminen on suoritettava erikseen “gem install <nimi>”-komennolla. Osa gemeistä vaatii vielä ruby-paketteja, kuten “ruby-devel”.

Lopuksi ajan skriptin nimeltä install_gems joka käytännössä asentaa OpenNebulan vaatimia kirjastoja.

Asennus onnistui, sillä OpenNebulan palvelu on tavoitettavissa selaimen kautta osoitteesta <http://<Front-End-IP>:9869>.

3.2.3 OpenNebula Node

Front-end tarvitsee noden, jossa on KVM ja libvirt asennettuna. Ne voi asentaa itse tai asentamalla OpenNebulan Node-paketit.

Jotta KVM toimisi OpenNebulan kanssa, niin siellä täytyy olla konfiguroituna oneadmin-käyttäjä salasanattomalla SSH-yhteydellä. Käytännössä se toteutetaan UID- ja GID-tietojen kopioimisella ja `"/.ssh/config"`-tiedoston muokkaamisella. Lisäksi ruby täytyy myös olla asennettuna. Kyseiset toimenpiteet tein ennen varsinaisen asennuksen alkua. Seuraavaksi aloitan libvirtin konfiguroinnin OpenNebulalle sopivaksi.

Aluksi avaan tekstieditorilla `/etc/libvirt/qemu.conf`-tiedoston ja vaihdan `dynamic_ownership`-arvoksi `"0"`. Vaihdan myös arvot `"user = oneadmin"` ja `"group = oneadmin"` OpenNebulan ohjeen mukaisesti. Yritettyäni KVM1:n lisäämistä OpenNebulaan saan jälleen virheen aikaiseksi. Tällä kertaa `kvm.rb` oikeudet eivät riitä suorittamiseen, joten etsin apua OpenNebulan sivuilta ja löydän sieltä apua ongelmaan käyttäjältä "Idafen". Luettuani keskustelun, lisään vielä `libvirtd.conf`-tiedostoon `"unix_sock_group = "oneadmin"'"` ja `"unix_sock_rww_perms = "0777"'"` Samalla valmistelen palvelimen käyttämään libvirtin migraatio-ominaisuuksia muokkaamalla `"/etc/libvirt/libvirtd.conf"`-tiedostoon `"listen_tcp = 1"`. Muutoksien jälkeen KVM-palvelin on hyvä käynnistää uudelleen, jotta tehdyt muutokset tulevat voimaan. (Lists.opennebula.org 2011)

Seuraavaksi siirryn Sunstone-käyttöliittymään lisäämään uuden KVM-isännän ympäristöön. Infrastructure-valikon alta löydän Hosts-kohdan. Create-painike avaa ikkunan, jossa kysytään tietoja hypervisor-hostista. Minun tapauksessani lisään KVM-hostin `192.168.200.11`-osoitteesta ja käytän KVM-ajureita. Määritän myös clusterin nimeltä "Cluster".

Pieni hetki lisäämisen jälkeen Sunstone alkaa monitoroimaan uutta KVM:ää ilman suurempia ongelmia. OpenNebula löytää KVM:stä rautatason tietoja ja piirtää grafiikkaa prosessorin käyttötasosta sekä muistin kulutuksesta.

3.3 Käyttöliittymät

OpenNebulassa hypervisoreita hallitaan joko komentorivi- tai web-pohjaisella käyttöliittymällä.

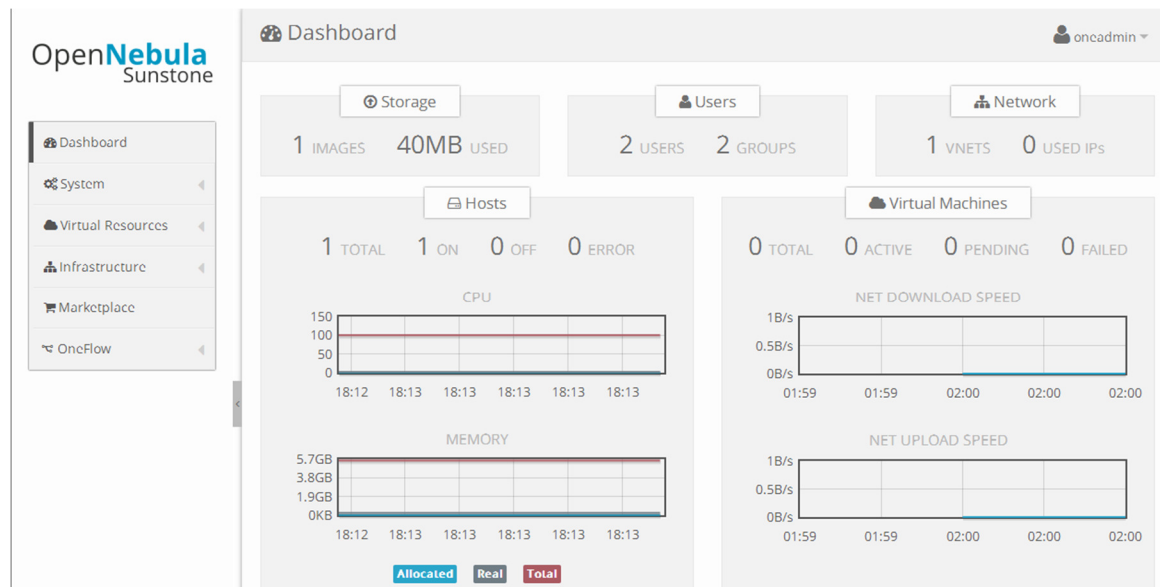
3.3.1 Komentorivi

CLI-komentoja käytettäessä ei tule sen enempää vaihtoehtoja kuin Suntonessakaan ympäristön konfigurointiin. Omassa käyttökokemuksessani CLI:n hyöty oli lähinnä joustava polkujen määrittäminen verrattuna Sunstoneen, kun latsin levykuva OpenNebulaan. Sunstonen kautta levykuva olisi pitänyt ladata paikalliselta tietokoneelta, kun taas komentorivillä mahdollisuuksia on muualtakin.

Käytännössä komentorivin kautta voi tehdä kaikki samat asiat kuin Sunstonesta, mutta sitä kautta ei myöskään saa enempää ominaisuuksia käyttöönsä. Virtuaalikoneiden luominen, verkkojen-, käyttäjien, ryhmien, levyjen, hypervisoreiden ja levykuvien hallinta on myös saatavilla sieltäkin.

3.3.2 Sunstone

Sunstone on web-käyttöliittymä OpenNebulaan. Oletuksena Sunstonen osoite on <http://<Front-End-IP>:9869>. Kuvassa 3 esitetään Sunstone-käyttöliittymän näkymä OpenNebulan virtualisointiympäristöön.



Kuva 3. Sunstone-käyttöliittymässä näkyy ympäristön tietoja.

Sunstonen etuna on graafinen näkymä hostien resursseihin. Siihen on pääsy kaikilta tietokoneilta, joissa on nykyaikainen Internet-selain ja jotka ovat samassa verkossa. Suntonessa on

myös integroitu marketplace, josta käyttäjät voivat ladata virtuaalikoneita tai levykuvia. Käyttöliittymä toimii nopeasti ja sivut päivittyvät automaattisesti, jotta käyttäjä on aina ajan tasalla.

Toisaalta Sunstone-käyttöliittymä sisältää ikäviä bugeja, joiden takia käyttökokemus saattaa kärsiä. Esimerkiksi uuden datastoren lisääminen Sunstonen kautta on hyvin buginen. Sunstone ei näytä aina kaikkia kenttiä, ellei valikoita selaa edestakaisin. Esimerkiksi LVM-kohdasta jää HOST- ja VG_name-kenttä pois, mikäli käyttäjä on jo lisännyt LVM-jaon ja haluaa lisätä heti perään toisen. Tällöin Sunstone muistaa, että käyttäjä loi viimeksi LVM-jaon ja näyttää sen oletuksena, mutta siitä puuttuu kyseiset kentät.

Toinen bugi datastoren lisäämisessä on se, että käyttäjä syöttää esimerkiksi LVM-kenttään tiedot HOST ja VG_name, mutta päättääkin tehdä filesystem-muotoisen datastoren. Tällöin pyyhkimättä jääneet kentät tulevat myös konfiguraatioon mukaan, vaikka niitä ei olisi haluttu.

Salasanan vaihtaminen Sunstonen kautta ei vaihda salasanaa Front-End:n one_auth-nimiseen tiedostoon, jota OpenNebula käyttää käyttäjän autentikoimiseen suorittaessa komentoja. Projektissani huomasin tämän, kun yritin listata virtuaalikoneita komentoriviltä onevm-komennolla. Tuloksena sain logi-tiedostoon merkinnän “VirtualMachinePoolInfo result FAILURE [VirtualMachinePoolInfo] User couldn't be authenticated, aborting call.” Vaihdamalla salasanan samaksi kuin Sunstonessa komennot rupesivat toimimaan oikein ja salasana oli sittemmin sama CLI:ssä ja Sunstonessa.

3.4 OpenNebulan ominaisuudet

Käyn läpi OpenNebulan tarjoamia ominaisuuksia Sunstone-käyttöliittymästä. Ominaisuudet olivat käytettävissä omassa ympäristössäni.

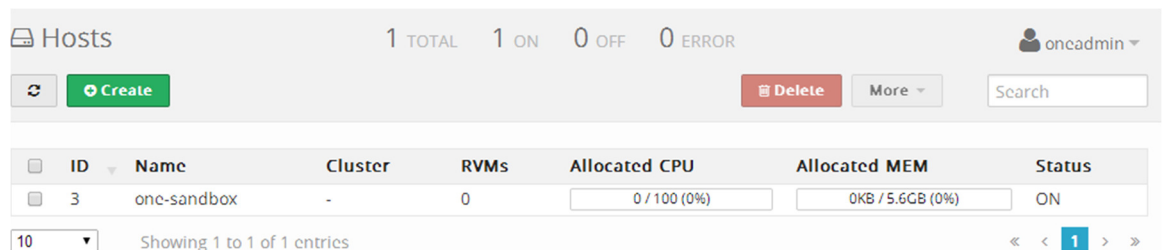
3.4.1 Klusteri

Klustereiden luomisella on mahdollista saavuttaa vakaa virtualisointi infrastruktuuri. OpenNebula tukee kyseistä ominaisuutta ja sen käyttäminen on helppoa. Klusterin luomisessa on mahdollista jakaa klustereille omat verkot, datastoret ja isännät. Käytännössä tällä ominai-

suudella voi luoda oman produktiopoolin, jossa on vaikkapa DNS, DHCP ja LDAP eristettyinä muiden käyttäjien virtuaalikoneista.

3.4.2 Hostit

OpenNebula tukee Xen, KVM, VMware ja EC2 hypervisoreita. Hypervisorit tarvitsevat kuitenkin suhteellisen paljon konfigurointia, ennen kuin ne toimivat täydellä teholla. Käytännössä se vaatii muutamien konfigurointi-tiedostojen muokkaamista ja salasanan SSH-yhteyden front-endistä KVM:lle. Sen jälkeen ne on mahdollista yhdistää Sunstonen kautta IP-osoitetta tai Hostnamea käyttämällä OpenNebulaan vaivattomasti (kuva 4).



ID	Name	Cluster	RVMS	Allocated CPU	Allocated MEM	Status
3	one-sandbox	-	0	0 / 100 (0%)	0KB / 5.6GB (0%)	ON

Kuva 4. OpenNebulan hostit näkyvät listattuna.

3.4.3 Datastore

Datastore on yksi suurimmista ongelmista OpenNebulassa. OpenNebula haluaa kontrolloida datastorea hyvin paljon, joka ei ole ideaalitalanne etenkin iSCSI-datastorelle. Jotta datastoren saa toimimaan, pitää väliin asentaa jokin Linux distributio ja luoda oneadmin-käyttäjä salasananomalla SSH-yhteydellä. Lisäksi oneadmin-käyttäjälle pitää luoda sudo oikeudet salasananomasti joihinkin komentoihin. Tämäkään ei takaa, että datastore menee saumattomasti OpenNebulaan, vaan datastorelle täytyy valita yhteystyyppi, jota pitää konfiguroida sen mukaan, millainen datastore on kyseessä.

Omassa asennuksessa ongelma nousee LVM-datastoren ja OpenNebulan välinen yhteys. OpenNebula ei missään vaiheessa ymmärtänyt sitä, kuinka paljon levytilaa LVM:ssä oli. Tästä syystä sinne ei voinut luoda yhtään tiedostoa, vaikka datastore liittyi OpenNebulaan hyvin.

Myös OpenNebulan logi sai mielestään hyvin informaatiota datastoresta vaikka tosiasiaassa se ei toiminut.

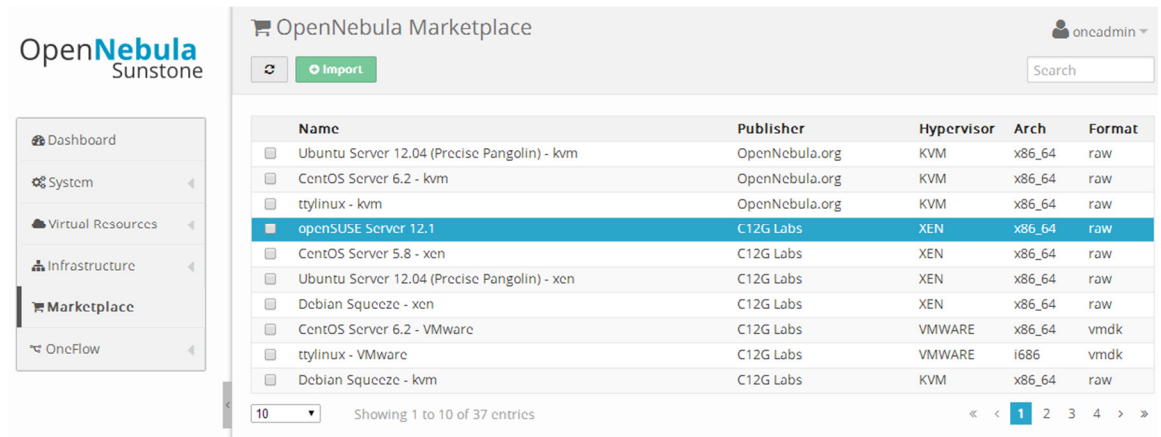
Datastoren tyyppi saattaa vaikuttaa siihen, kuinka hyvin OpenNebula ottaa haltuunsa levyjärjestelmän. LVM:n kohdalla se ei kovin hyvin toiminut, mutta paikallinen levy toimi loistavasti. Muita tuettuja tekniikoita ovat iSCSI, Ceph ja VMware VMFS.

3.4.4 Verkot

OpenNebulassa on mahdollista määrittää virtuaalisia verkkoja, joita virtuaalikoneet käyttävät. Virtuaalikoneet saavat IP-osoitteen jo käynnistysvaiheessa varatuksi ja niitä voi katsella Virtual Networks -valikosta, jossa näkyy lista osoitteista ja virtuaalikoneista jotka niitä käyttävät.

3.4.5 Marketplace

Marketplace on paikka, josta käyttäjät voivat ladata levykuvia, virtuaalikoneita, virtuaalilaitteita tai työkaluja (kuva 5). Lokakuussa 2013 OpenNebula Marketplacesta löytyy 28 kohdetta. Käytännössä kohteet eivät ole levykuvia käyttöjärjestelmistä niiden tavanomaisessa muodossaan, vaan ne on konfiguroitu toimivaksi OpenNebulassa. Rajoituksena kohteissa on vielä sekin, että kaikki eivät toimi kaikissa ympäristöissä. Tämä johtuu siitä, että OpenNebula tukee useaa eri hypervisoria, mutta silti kaikki hypervisorit eivät toimi samalla tavalla. Tästä syystä Marketplacesta tulee katsoa, mille hypervisorille kohde on tehty ja mitä hypervisoria itse käyttää.

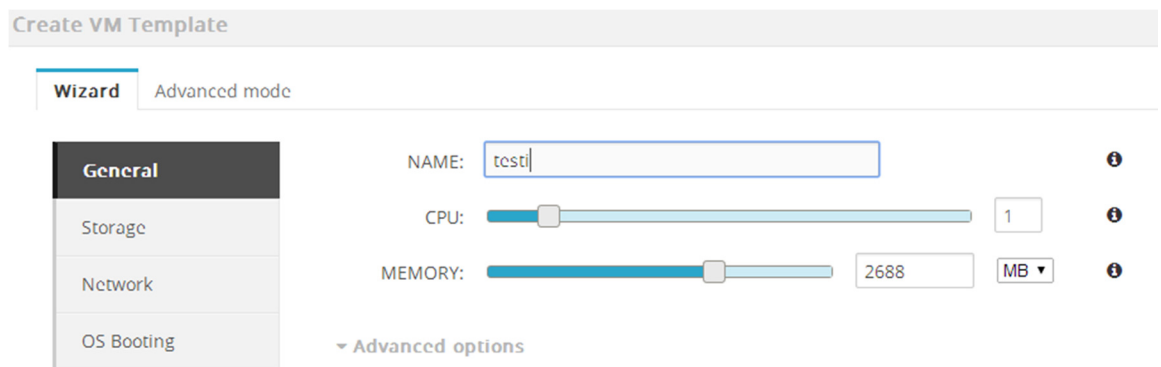


Kuva 5. OpenNebula Marketplace tarjoaa levykuvia käyttäjille.

Marketplacen palvelimet toimivat poikkeuksetta nopeasti ja jo muutamassa minuutissa virtuaalikoneen saa jo toimintakuntoon. Lisäksi Marketplacessa kaikki on ilmaista.

3.4.6 Template

Ennen virtuaalikoneen luomista luodaan template (kuva 6). Jo templatien tekemisessä määritetään virtuaalikoneen resurssit prosessorista ja kovalevystä lähtien. Toimivalle templatelle OS Booting -valikko on erittäin tärkeä. Sieltä määritetään prosessorin arkkitehtuuri, joka on oletuksena i686 eli 32-bittinen. Myös käynnistystapa on hyvä valita tehdyn levykuvan mukaiseksi. Levykuva liitetään Storage-valikossa riippumatta siitä, minkä tyyppistä levykuvaa ollaan liittämässä. Asetukset ovat päivitettävissä luomisen jälkeenkin.



Kuva 6. Virtuaalikoneen template luodaan ennen virtuaalikoneen käyttöä.

Virtuaalikoneen käytön helpottamiseksi näyttötekniikka valitaan myös templatien tekemisen aikaan. Se on välttämätöntä, mikäli käyttäjä haluaa päästä virtuaalikoneeseen käsiksi. Oletuksena virtuaalikoneet käyttävät VNC-tekniikkaa ja kuuntelevat kaikkia osoitteita arvolla 0.0.0.0. Myös SPICE-tekniikka on tuettu.

3.4.7 Levykuvat

Levykuvat voivat olla hyvin monenlaisia. Onnistunut virtuaalikoneen luominen vaatii tietämystä siitä, millaisessa muodossa levykuva on. Esimerkiksi Marketplace tarjoaa valmiita virtuaalikoneita, joiden tyyppi on OS, kun taas tavallisen distribuution levykuva on usein CDROM. Väärän tyypin määrittäminen johtaa virtuaalikoneeseen, joka ei käynnisty.

3.4.8 OneFlow

Oneflow on ominaisuus, jolla käyttäjä voi määrittää ja aikatauluttaa palveluiden roolit ja toiminnot niissä tilanteissa, kun järjestelmässä tapahtuu joko sammutus tai käynnistys operaatio (kuva 7). (OpenNebula c 2014)

The screenshot shows the 'Create Service Template' dialog box in the OpenNebula OneFlow - Templates interface. The dialog has a sidebar on the left with a menu containing 'Dashboard', 'System', 'Virtual Res', 'Infrastruct', 'Marketplac', 'OneFlow', 'Services', and 'Templates'. The main area contains the following fields and sections:

- Name:** A text input field.
- Strategy:** A dropdown menu with 'Straight' selected.
- Shutdown action:** A dropdown menu.
- + ADD ANOTHER ROLE** button and **Role 0** tab.
- Role Name:** A text input field.
- VM template:** A dropdown menu with 'ttylinux (id:0)' selected.
- Cardinality:** A text input field with '1'.
- Shutdown action:** A dropdown menu.
- Elasticity -** section:
 - Min VMs:** A text input field.
 - Max VMs:** A text input field.
 - Cooldown:** A text input field.
 - Elasticity policies** table:

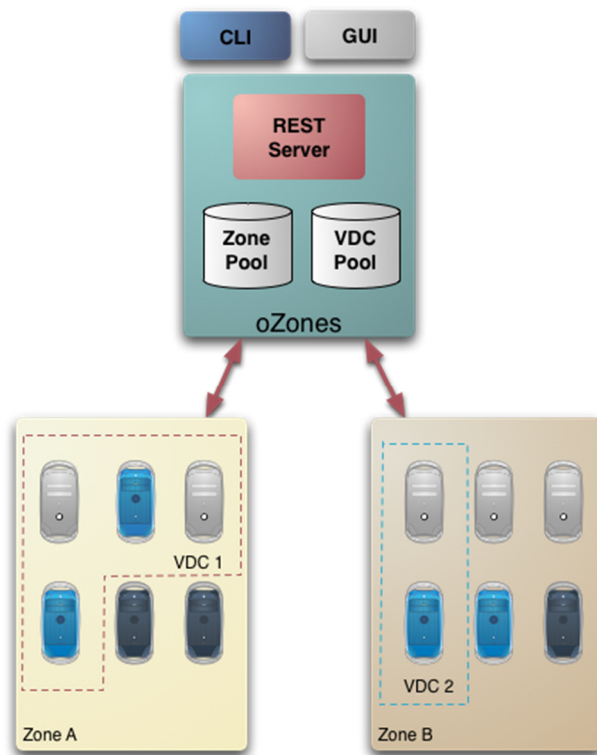
Type	Adjust	Min	Expression	# Periods	Period	Cooldown
+ Add						
 - Scheduled policies** table:

Type	Adjust	Min	Time format	Time expression
+ Add				
- Reset** and **Close** buttons at the bottom left.
- Create** button at the bottom right.

Kuva 7. Ylläpitäjällä on mahdollisuus hallita Oneflow-tapahtumia

3.4.9 Virtuaalinen datacenter

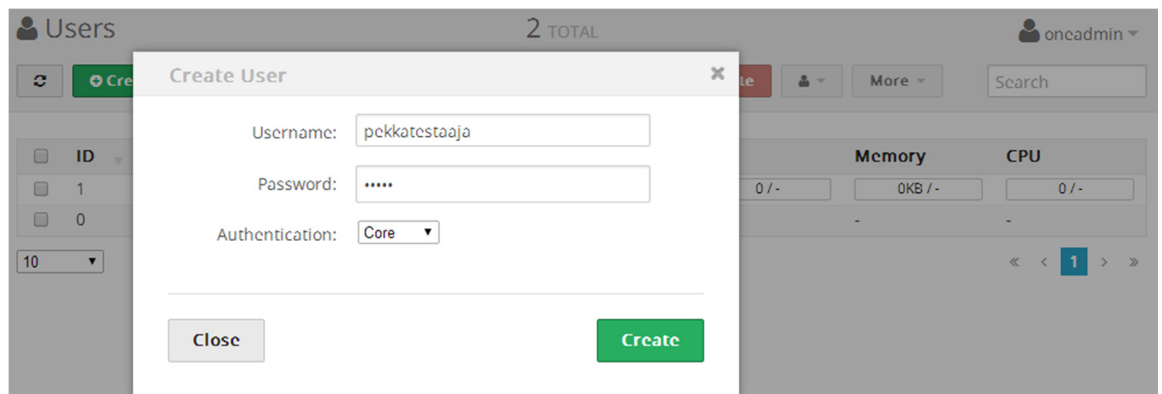
OpenNebula Zones on palvelu, joka mahdollistaa keskitetyn hallinnan OpenNebulan instansseille, jotka ovat esimerkiksi eri domaineissa (kuva 8). Sillä hallinnoidaan virtuaalisia datacenter-resurssipalasia, jolloin eri alueet voidaan eristää toisistaan. Käytännön hyöty virtuaalisille datacentereille tulee siitä, että käyttäjiä ja yrityksiä voidaan eristää omiin alueisiinsa eri asetuksilla. OpenNebula Zonesin Web-hallintapaneelin kautta voidaan luoda virtuaalisia datacentereitä. Kun datacenter luotu, niin se on saatavilla osoitteessa <http://<OpenNebula IP>/<Oma datacenter>>. Zones on saatavilla EPEL-repositoriosta nimellä `opennebula-ozones`. (OpenNebula d 2014)



Kuva 8. Visuaalinen kuvaus OpenNebula Zones-palvelusta (OpenNebula d 2014)

3.4.10 Käyttäjien hallinta

Käyttäjän luomisessa määritetään tapa, jolla käyttäjä autentikoidaan. Esimerkiksi Core-autentikointi etsii Linux-käyttäjää (kuva 9). LDAP käyttäjätietokanta voidaan myös perustaa ja liittää OpenNebulaan. Resurssien käyttöä voidaan seurata käyttäjäkohtaisesti.



Kuva 9. Käyttäjän luominen on helppoa

Käyttäjiä voidaan jakaa ryhmiin, jotta heidän hallitsemisensa olisi helpompaa. Pääsynvalvon-
nalla ryhmille määritetään resurssit, joihin ne pääsevät käsiksi. Sellaisia resursseja voivat olla
dokumentit, Hostit eli isännät, virtuaalikoneet, virtuaaliverkot, levykuvat ja templatet. Pääsyä
voidaan joko rajoittaa, sallia tai estää.

3.5 Virtuaalikoneen luominen

Valmiista templatesta voidaan luoda kerralla useita virtuaalikoneita. Luominen on hyvin yk-
sinkertaista, jos templatet on luotu oikein. Mikäli templatessa on määritetty NIC eli verkko-
kortti sekä virtuaaliverkko, niin virtuaalikone saa automaattisesti IP-osoitteen.

3.5.1 Levykuvan tuonti

Levykuvan tuonti tarkoittaa tässä yhteydessä levykuvan liittämistä OpenNebulan ympäris-
töön. Levykuvan tuominen alkaa Images-valikosta.

Uuden levykuvan tyypiksi määritetään CD-ROM, sillä kyseessä on asennusmedia. Datasto-
reksi valitaan jokin siihen hyvin soveltuva vaihtoehto oman ympäristön mukaisesti. Itse käy-
tin LVM-datastorea, joka sijaitsee eräällä KVM-hostilla. Poluksi asetetaan levykuvan lataus-
osoite. Esimerkiksi `"http://torvi.atm.tut.fi/ubuntu-releases/12.04.4/ubuntu-12.04.4-
desktop-amd64.iso"`, sillä siellä sijaitsee Ubuntu 12.04:n levykuva. OpenNebulan grafiikassa
statuksena lukee "LOCKED", kunnes levykuva on ladattu kokonaan.

3.5.2 Levyaseman luominen

Uusi virtuaalikone tarvitsee tyhjän virtuaalilevyn, jolle juuri luotu asennusmedia asennetaan.
Uuden levyn määrittämisessä käytetään samaa Images-välilehteä kuin aiemmin. Levykuvan läh-
teeksi kelpaa paikallinen tai verkosta löytyvä ISO-levykuva (kuva 10).

Image location:

☒ Provide a path
 ☐ Upload
 ☐ Empty datablock

Path: ⓘ

Kuva 10. Levykuvan lähde syötetään kenttään.

Mikäli käyttäjä haluaa asentaa CDROM-tyyppisen levykuvan, on hyvä olla ainakin yksi Volatile Disk eli tyhjä levy liitettynä levykuvan lisäksi. Volatile Disk/Empty Datablock paljastaa kaksi kenttää "Size" ja "FS type". Size-kohtaan laitetaan tallennustilan koko ja FS type-kenttään lisätään käyttöjärjestelmän tukeva tiedostojärjestelmä, esimerkiksi "ext2". Kun uusi levy näkyy listassa, niin sen tyyppi täytyy muuttaa muotoon OS ja laitetaan ruksi kohtaan "Persistent", jotta käyttöjärjestelmä voidaan asentaa sille.

3.5.3 Templaten luominen

Templates-valikosta täytyy käydä luomassa uusi template, josta sitten käännetään virtuaalikone.

Templatelle valitaan asennusvelho välilehdistä, joista käytössä KVM-ympäristön mukaisesti on "Wizard KVM". Resurssit voidaan jakaa sen mukaan, mitä ympäristössä on käytössä. Boot/OS options -valikossa arkkitehtuuriksi on valittava joko 32- tai 64-bittinen riippuen siitä, kumpaa käyttöjärjestelmää ladattava levykuva on. Boot-kohdassa määritetään se, mistä virtuaalikone käynnistyy. CD-levy kuvaa käytettäessä käynnistetään myös virtuaalikone CDROM-vaihtoehdolla. Valmiiksi luotu CD-ROM levykuva ladataan Add disks/images-kohdasta ja sama toimenpide tehdään OS-tyyppiselle virtuaalilevylle. Lopuksi lisätään Add Graphics -valikosta VNC-tyyppi, jotta saadaan kuvayhteys virtuaalikoneeseen.

3.5.4 Virtuaalikone

Uuden virtuaalikoneen voi luoda painamalla Instantiate-painiketta Template-valikossa tai Create-painiketta Virtual Machines -valikossa. Luomistyökalu ilmestyy ruutuun, jolloin käyttäjä voi valita templatien ja kopioiden määrän (kuva 11).

Create Virtual Machine

Step 1: Specify a name and the number of instances

VM Name: ⓘ # VMs: ⓘ

Step 2: Select a template

ID	Name
0	ttylinux

« < 1 > »

You selected the following template: **ttylinux**

Kuva 11. Virtuaalikoneita kopioidaan templatesta.

Vamista virtuaalikonetta voi muokata rajoitetusti. Käytännössä käyttäjä voi muuttaa vain le-
vyasemia ja verkkokortteja. Virtuaalikoneista voi ottaa myös ”snapshotteja” ja niitä voi mig-
ratoida hostilta toiselle.

3.6 Teknologia

OpenNebula käyttää hyödykseen libvirt-teknologiaa. OpenNebula on ns. pilvialusta joka tu-
kee yksityis- ja hybrid-ratkaisuja, joissa voidaan käyttää omaa infrastruktuuria yksin tai yhdes-
sä julkisen pilvipalvelun, esimerkiksi AWS:n, kanssa. (OpenNebula e 2014)

OpenNebulan suunnittelun kulmakivi on avoimuus, jossa yhteisö pääsee halutessaan käsiksi
koodiin. Sitä voidaan käyttää missä tahansa datacenterissä, ja sillä voidaan tavoitella yritysta-

son palvelua ympäristössä, jota ylläpitäjä hallitsee itse. Siitä yritetään luoda mahdollisimman yksinkertaista käyttäjälle ja kevyttä IT-ympäristölle. (OpenNebula f 2014)

4 PROXMOX VE 3.1

Proxmox Server Solutions GmbH on Itävallassa sijaitseva yritys, joka kehittää Proxmox Virtual Environment- ja Proxmox Mail Gateway -palveluita. Se on ollut toiminnassa vuodesta 2004 saakka. (Proxmox b 2014)

4.1 Rakenne

Proxmox rakentuu samasta levykuvasta asennettavasta ohjelmistosta. Front-end ja node kasataan samanlaisesta asennuksesta, mutta ne sitten liitetään keskenään yhteen. Proxmox ei seikoita asennusta erillisillä asennuspaketeilla ja riippuvuuksilla, vaan kaikki kuuluvat samaan kokonaisuuteen. Tällöin esimerkiksi web-palvelin on kaikissa Proxmox VE -asennuksissa mukana. Käytännössä hallintanäkymät yhdistyvät samaan käyttöliittymään.

4.2 Proxmox VE:n asennus

Proxmox.com tarjoaa etusivullaan (<https://www.proxmox.com/>) valmista levykuvaa Proxmox 3.1-versiosta. Lataan sen ja asennan palvelimeen, jossa on 4 GB muistia, yksi prosessori ja 16 GB levytilaa.

Proxmox on periaatteessa Debianin pohjalle asennettu valmis palvelin, jossa on KVM integroituna. Asennuksen alussa määritän vain salasanan root-käyttäjälle ja IP-asetukset.

Asennuksen jälkeisen uudelleen käynnistytksen jälkeen Proxmox on jo käyttövalmis. Proxmoxissa web-palvelin käynnistyy ja pääsen siihen käsiksi osoitteesta <https://<proxmox-IP>:8006>. Palvelin on valmis käyttöä varten.

4.3 Klusteri

Klusterin lisääminen Proxmoxissa onnistuu komentoriviltä. Vaatimuksena on vähintään kahden Proxmox VE -palvelimen asentaminen. Kun kaksi palvelinta (PVE1 ja PVE2) on asennettu, edetään toisen asennettavan komentoriville. (Proxmox c 2014)

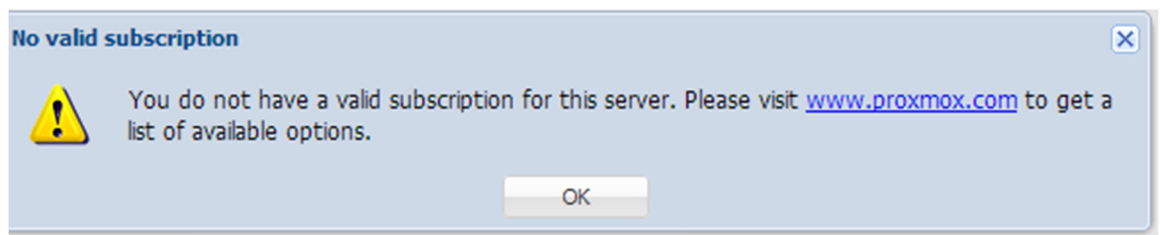
PVE1-palvelimella avataan komentorivi, jossa luodaan klusteri komennolla "pvecm create <klusterin nimi>". Klusterin luomisen jälkeen lisätään node PVE2 klusteriin PVE2:n komentoriviltä komennolla "pvecm add <PVE1-IP>".

4.4 Subscription

Proxmox tarjoaa lisäpalveluita subscriptionin omistajille. Hinta vaihtelee neljän eri tason mukaan, kuten myös ominaisuudet. Subscriptionissa käyttäjä saa pääsyn yhteisöfoorumille, vakaita päivityksiä, tukipyyntöjä, etäyhteyden SSH:n kautta, pääsyn asiakasportaaliin ja Enterprise repositoryyn.

Ilmaiskäyttäjälle subscriptionin puute näkyy kirjautumisruudun jälkeen, kun ruutuun tulee ilmoitus (kuva 12). Kyseisen ilmoituksen saa pois epävirallisella korjauksella "wget

`http://www.tolaris.com/blog/wp-content/uploads/2013/08/no_subscription_popup_pvemanagelib.js-3.1.patch -O - | patch -p0"` (Tolaris.com 2013)



Kuva 12. Käyttäjälle ilmoitetaan subscriptionin puute.

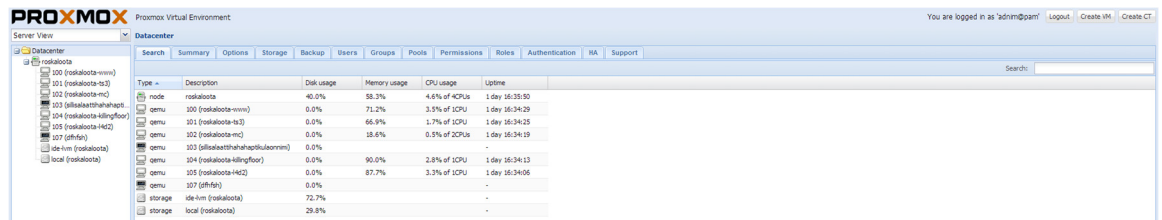
Patchin jälkeen tulee käynnistää palvelin uudestaan, jotta muutokset tulevat voimaan. Sama operaatio tulee toistaa kaikissa palvelimissa, joissa käytetään web-käyttöliittymää. Korjaus ei toimi, jos Proxmoxin päivittää.

4.5 Proxmoxin ominaisuudet

Käyn seuraavassa osiossa läpi Proxmox VE:n ominaisuuksia. Konfiguroin kyseisiä ominaisuuksia samalla toimintakuntoon tarpeen mukaan.

4.5.1 Käyttöliittymä

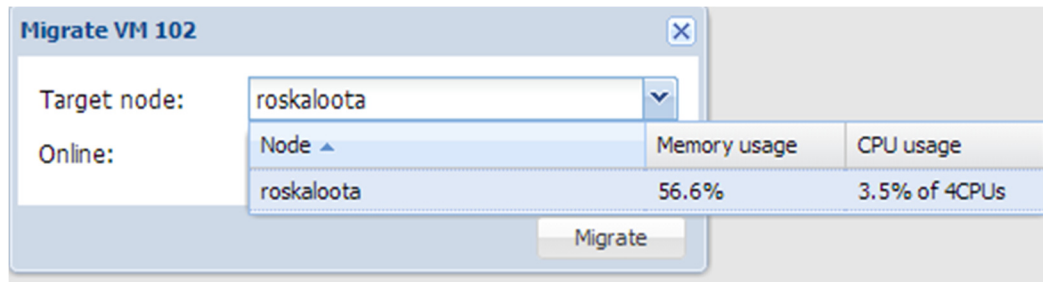
Proxmoxin web-käyttöliittymä erottuu muista sillä, että se näyttää ympäristön hierarkianäkymää muistuttavalla tavalla. Lisäksi valittavasta kohteesta saa lisää valikoita käyttämällä välilehtiä (kuva 13).



Kuva 13. Proxmox erottuu muista käyttöliittymällään.

4.5.2 Migraatio

Proxmox sisältää migraatio- ja live-migraatio-ominaisuuden (kuva 14). Jotta Live-migraatio toimisi, täytyy kovalevyn formaatin olla joko IDE tai Virtio. Muutoin Proxmox ilmoittaa virheellisestä laitteesta. Muutoin se toimii jo ilman mitään konfigurointeja. Tavallinen migraatio ja tietokoneen kloonaukset toimivat myös ilman ongelmia. Käyttäjä ei huomaa virtuaalikoneen siirtymistä KVM-hostilta toiselle live-migraatiota käytettäessä.



Kuva 14. Virtuaalikoneen migraatio onnistuu valitsemalla palvelimen.

4.5.3 Käyttäjien hallinta

Käyttäjien autentikointi toimii joko LDAP-, Active Directory-, Proxmox VE- tai Linuxin oman autentikointitekniikan kautta. Oletuksena Proxmox käyttää Linuxin ja Proxmoxin omaa tekniikkaa. Ylläpitäjä saa itse valita, millä käyttäjä luodaan.

Ylläpitäjä voi luoda käyttäjiä datacenter-valikon users-välilehdeltä. Käyttäjän luomisessa voi määrittää yhteystiedot, käyttäjätunnukset, ryhmät ja sen, minne käyttäjä tallennetaan (kuva 15). Mikäli ylläpitäjä haluaa luoda väliaikaisen tunnuksen esimerkiksi kahdeksi viikoksi, niin sekin onnistuu.

Kuva 15. Käyttäjille on olemassa muitakin tietoja kuin pelkkä käyttäjänimi.

Luomalla ryhmän, ylläpitäjä voi erottaa käyttäjät toisistaan ja laittaa kommentin ryhmälle. Ryhmän tekeminen ei itsessään rajoita pääsyä mihinkään, ennen kuin sääntöjä on luotu.

Pääsynvalvonnassa ylläpitäjä voi luoda sääntöjä yksittäiselle käyttäjälle tai ryhmälle. Säännöissä on jo oletuksena erilaisia tasoja datacenterin ominaisuuksiin. Proxmox osaa myös piilottaa resurssit, joihin käyttäjällä ei ole pääsyä.

4.5.4 Autentikointipalvelin

Proxmox VE on itsessään oma autentikointipalvelin. Se perustuu Unixin omaan autentikointijärjestelmään. Proxmox VE varastoi salasanat SHA-256-salauksella `/etc/pve/priv/shadow.cfg`-tiedostoon. Käyttäjätiedot ovat puolestaan `/etc/pve/user.cfg`-tiedostossa. Käyttäjien hallinta onnistuu web-käyttöliittymässä tai komentoriviltä Proxmox VE User Management -ohjelmalla. Ohjelman käyttäminen onnistuu komennolla `pveum`. Se asentuu Proxmoxin asennuksen yhteydessä. (Proxmox d 2014)

4.6 High Availability

Proxmoxissa on myös High Availability -mahdollisuus. Käytännössä se tarkoittaa sitä, että vaikka ympäristössä tapahtuisi esimerkiksi laitteiston hajoaminen, niin datacenter ympäristö jatkaa toimintaansa.

4.6.1 Two-Node Availability

Proxmoxin "Two-Node High Availability" tarkoittaa HA-ominaisuuksia kahdelle nodelle. Dokumentaatio osoitteessa "http://pve.proxmox.com/wiki/Two-Node_High_Availability_Cluster" suosittelee vähintään kolmea nodea HA-ympäristön tekemiseen, mutta se onnistuu myös kahdella nodella. Kahden noden HA:n säätäminen vaatii pientä lisää verrattuna kolmen tai useamman noden konfigurointiin. Dokumentaation mukaan outouksia saattaa tapahtua, kun nodeja on vähemmän kuin kolme.

4.6.2 Fencing

Ympäristössäni on kaksi nodea ja kummatkin tarvitsevat samat toimenpiteet. Asennukseen käytän apuna virallisia dokumentaatioita osoitteissa: "http://pve.proxmox.com/wiki/Two-Node_High_Availability_Cluster" ja "<http://pve.proxmox.com/wiki/Fencing>".

Aloittaakseni asennuksen kirjaudun sisään Node1:lle ja editoin komennolla `"nano /etc/default/redhat-cluster-pve"` FENCE_JOIN arvoksi `"yes"`. Lisäksi yhdistän Node1:n fence domainiin komennolla `"fence_tool join"`. Käytännössä otin fencingin käyttöön Node1:llä.

Seuraavaksi luon kopion cluster.conf-tiedostosta komennolla `"cp /etc/pve/cluster.conf /etc/pve/cluster.conf.new"`. Kyseiseen tiedostoon editoidaan fence-laitteet. Editoidessa on hyvä muistaa manuaalinen editointi tiedoston versionumeroon, jotta Proxmox osaa päivittää tiedoston uusimpaan versioon.

Määritän kyseiseen tiedostoon iDrac6-asetukset. Tarvittavat tiedot ovat käyttäjänimi, salasana ja IP-osoite, josta iDrac löytyy.

Seuraavaksi siirryn takaisin web-käyttöliittymään, jossa HA-valikko ilmoittaa seuraavaa: `"mismatched tag at line 21, column 2, byte 524 at /usr/lib/perl5/XML/Parser.pm line 187 (500)"`. Kyseinen herja tarkoittaa sitä, että konfiguroidussa tiedostossa on hyvin luultavasti kirjoitusvirhe. Mikäli kaikki meni, kuten oli tarkoitettu, niin käyttöliittymään pitäisi ilmestyä juuri konfiguroidut asetukset näkyviin (kuva 16).

Search	Summary	Options	Storage	Backup	Users	Groups	Pools	Permissions	Roles	Authentication	HA	Support
Add	Remove	Edit	Revert changes	Activate								
Tag	Attributes											
cluster	config_version="4" name="proxmox-cluster"											
cluster	expected_votes="1" keyfile="/var/lib/pve-cluster/corosync.authkey" two_node="1"											
cluster	nodeid="1" name="kvm1" votes="1"											
cluster	name="1"											
cluster	action="reboot" name="node1-drac"											
cluster	nodeid="2" name="kvm3" votes="1"											
cluster	name="1"											
cluster	action="reboot" name="node2-drac"											
cluster	secure="1" name="node1-drac" login="root" passwd="██████████" agent="fence_drac5" cmd_prompt="admin1->" ipaddr="192.168.200.251"											
cluster	secure="1" name="node2-drac" login="root" passwd="██████████" agent="fence_drac5" cmd_prompt="admin1->" ipaddr="192.168.200.253"											

Kuva 16. HA-valikko näyttää konfiguroidut asetukset.

Testaan HA:n toiminnan luomalla uuden virtuaalikoneen, jossa optio “enable HA” on valittuna. Kyseinen optio löytyy HA-välilehden add-kohdasta. VM ID -kenttään kirjoitan arvon 100, joka on luomani virtuaalikoneen ID.

Seuraavaksi testaan HA:n toimivuutta kaatamalla toisen KVM-palvelimen. Testin aikana ilmenee kuitenkin virhe “kvm3 fence_drac5: Parse error: Ignoring unknown option 'no-dename=kvm1’”. Yrittäessäni etsiä apua internetistä huomasin, että kovin monella ei ole juuri samanlaista ongelmaa.

Hetken tutkimisen jälkeen kävi ilmi, että on olemassa toinenkin tekniikka nimeltään IPMI. Samasta Proxmox ohjeistuksesta löytyvällä ohjeella määritän periaatteessa samat asetukset kuin iDrac:illä, mutta ”fencing agent” vaihtuu fence_ipmilan-agentiksi.

Yrittäessäni uudestaan HA-testiä katkaisen virrat toisesta KVM-palvelimesta. Kyseisen toimenpiteen aikaan virtuaalikone ID:llä 100 siirtyy palvelimelta toiselle suhteellisen nopeasti. Noin vajaan minuutin käyttökatkon jälkeen virtuaalikone toimii kuten ennenkin.

Lopullisen HA-konfiguroinnin eli cluster.conf-tiedoston sisältö näyttää seuraavalta:

```
<?xml version="1.0"?>
<cluster config_version="10" name="proxmox-cluster">
  <cman expected_votes="1" keyfile="/var/lib/pve-cluster/corosync.authkey" two_node="1"/>
  <clusternodes>
    <clusternode name="kvm1" nodeid="1" votes="1">
      <fence>
        <method name="1">
          <device name="ipmi1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="kvm3" nodeid="2" votes="1">
      <fence>
        <method name="1">
          <device name="ipmi3"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_ipmilan" name="ipmi1" lanplus="1" ipaddr="192.168.200.251" login="root"
passwd="xxx" power_wait="5"/>
    <fencedevice agent="fence_ipmilan" name="ipmi3" lanplus="1" ipaddr="192.168.200.253" login="root"
passwd="xxx" power_wait="5"/>
  </fencedevices>
</rm>
<pvevm autostart="1" vmid="101"/>
</rm>
```

4.7 Datastore

Proxmox tukee useaa erilaista storage-tyyppiä. Nykyisessä ympäristössäni käytössä ovat iSCSI-, LVM-, Local- ja NFS-tyypin datastoret.

Datastoren lisääminen onnistuu Storage-välilehdeltä, josta näkee myös tämänhetkiset storage sekä niiden sisällön ja tyypin. Käytännössä lisäyskenttään laitetaan olemassa olevan datastoren osoite ja sen pitäisi toimia suoraan, ellei datastorea ole konfiguroitu väärin tai verkossa ole esteitä. Palvelimella olevat jaot tulevat näkyviin automaattisesti (kuva 17).

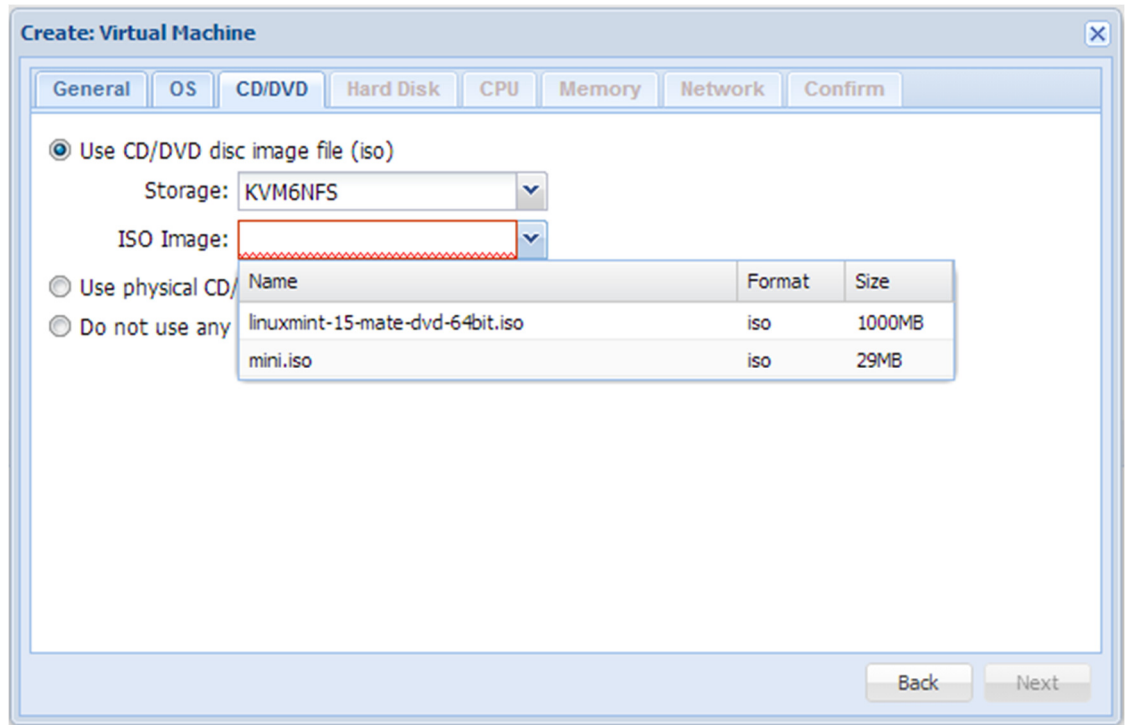
Add: NFS share

ID:	<input type="text"/>	Nodes:	All (No restrictions) ▼
Server:	192.168.200.19	Enable:	<input checked="" type="checkbox"/>
Export:	<input type="text"/>	Max Backups:	1 ▲▼
Content:	<div> /mnt/opennebula/Files /mnt/opennebula/System /mnt/opennebula/Images </div>		
<input type="button" value="Add"/>			

Kuva 17. Proxmox löytää automaattisesti NFS-jakoja palvelimelta.

4.8 Virtuaalikoneen luominen

Proxmoxissa luodaan uusi virtuaalikone painamalla Create VM -painiketta oikeassa ylänurkassa. Virtuaalikoneen luominen on hyvin yksinkertaista, sillä se ei vaadi templatien tekemistä. Riittää, että käyttäjältä löytyy levykuva datastoresta, jota käytetään virtuaalikoneen asennuksessa, fyysinen CD-levy tai valmiiksi asennettu virtuaalinen kovalevy (kuva 18). Hard-disk-välilehdellä vaaditaan myös storage, jonne virtuaalilevy tallennetaan. Bus/Device-valikossa valitaan sopiva väylä. Itse käytän VIRTIO-vaihtoehtoa, sillä se on osoittautunut toimivaksi Windows- ja Linux-ympäristöille. Network-välilehdellä sillattua tilaa käytettäessä täytyy valita myös verkkokortin tyyppi. Intel E1000 on toimiva ratkaisu Windowsilla sekä Linuxilla.



Kuva 18. Virtuaalikoneen luominen on tehty helpoksi.

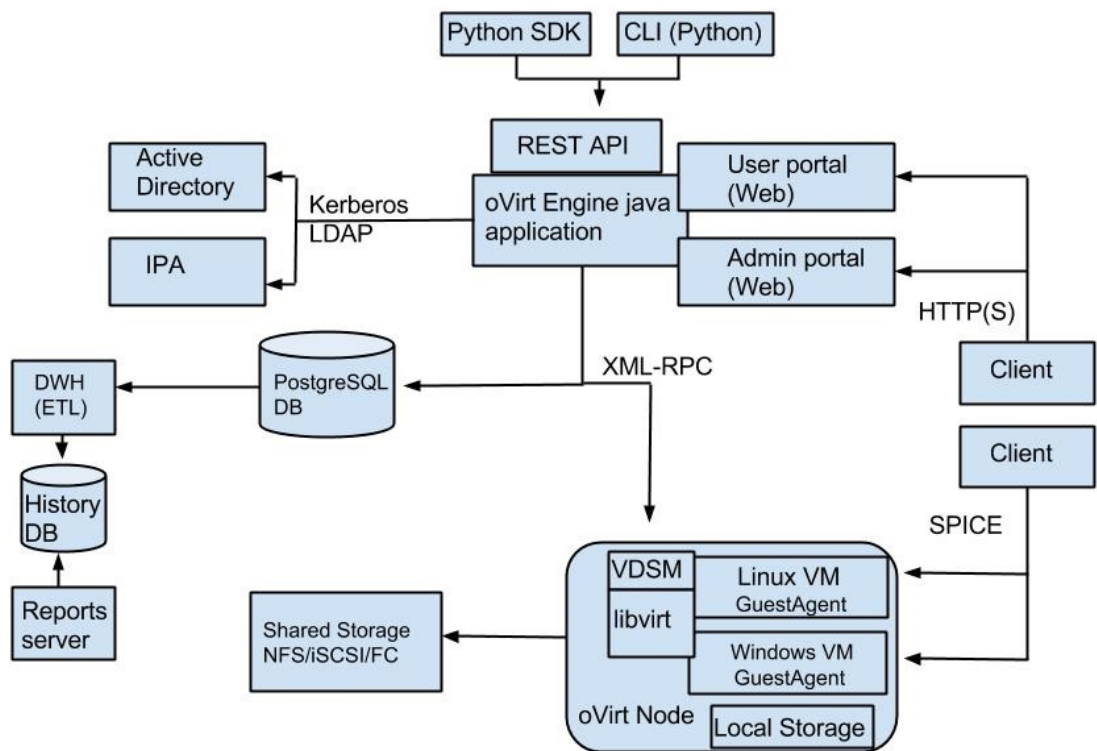
Valmista virtuaalikonetta voi muokata rajoitetusti. Hardware-välilehdessä virtuaalikoneelle voidaan lisätä laitteistoa ja muokata esimerkiksi kovalevyn kokoa, muistin määrää tai näppäimistön kieltä.

5 OVIRT 3.3

oVirtä kehittää yhteisö, joka kehottaa ihmisiä liittymään projektiin. Yhteisö on jaettu eri osa-alueisiin, jotka kehittävät tietynlaisia komponentteja suuremmasta kokonaisuudesta. (oVirt 2014)

5.1 Rakenne

oVirt koostuu oVirt Engine- ja oVirt Node -komponenteista. Lisäksi tarvitaan storage- ja verkkolaitteet (kuva 19).



Kuva 19. oVirtin rakennetta havainnollistava kuva.

5.1.1 oVirt Engine

oVirt Engine on Javaan perustuva ohjelma, jolla verkkopalvelut toimivat. Se on yhdistetty VDSM-palveluun. oVirt Enginen kautta tulevat sellaiset ominaisuudet, kuten LDAP-autentikaatio, storage, HA, levykuvan hallinta, migraatio ja load-balancing. Käytännössä se on front-end. (Lesovsky 2013, 11)

5.1.2 oVirt Node

oVirt Node on palvelin, joka tarjoaa tarvittavat resurssit oVirt Enginelle. Resurssien jakamisen mahdollistavat libvirt ja VDSM, jotka asennetaan oVirt Nodeen. Mikäli paikallista tallennustilaa käytetään, niin nodet tarjoavat sen. (Lesovsky 2013, 8)

5.2 Asentaminen

oVirtin asentamiseen on kaksi erilaista keinoa. Ensimmäinen vaihtoehto on ladata valmis oVirtin levykuva sisältäen kaiken tarvittavan. Toisessa vaihtoehdossa asennetaan ensin Linux, jonka päälle asennetaan oVirtin paketit. Ennen varsinaisen asennuksen alkua palomuurin tulee olla poissa päältä. Palomuurin voi tarvittaessa konfiguroida asennuksen jälkeen, kun ympäristö on todettu toimivaksi.

5.2.1 oVirt Enginen asennus

Löysin CentOS-asennusohjeen CentOS-wiki sivulta osoitteesta "<http://wiki.centos.org/HowTos/oVirt>". Sitä käyttämällä aloitan ovirt-enginen asennuksen. Samat ohjeet pätevät myös Fedoraan, mutta eri asennuspaketille repositorion puolelta.

Aluksi lisään Linuxiin EPEL-repositorion "https://dl.fedoraproject.org/pub/epel/6/x86_64/". Toisin kuin wikin ohjeessa, lisäsin EPEL-repositorion, koska siellä on uudempi versio oVirtistä kuin CentOS:lla omassa repositoriossaan. oVirt noudetaan repositoriosta komenolla "`yum install ovirt-engine`". Asennus

etenee ilman ongelmia ja oVirt on valmis ensimmäistä käynnistystä varten komennolla "engine-setup". Ajan setupin oletukseksi ehdotettavilla asetuksilla, sillä ne näyttävät sopivan ympäristölleni hyvin. Ne kohteet, jotka eivät ole asennusvaiheessa selvillä, voidaan konfiguroida myöhemmin uudelleen. Palvelut lähtevät käyntiin automaattisesti uudelleenkäynnistyksessä, jolloin hallintapaneliin pääsee osoitteella "http://<ovirt-enginen IP>:80/ovirt-engine".

5.2.2 Noden asennus

oVirt noden asennusohjeet löytyvät osoitteesta "http://www.ovirt.org/Quick_Start_Guide#Install_oVirt_Node". Käyttämässäni asennusohjeessa mainittu linkki ei toimi, sillä tiedostot ovat muuttuneet palvelimella. Oikeat paketit löytyvät selaamalla palvelimella olevia tiedostoja.

Asennuksen suoritan Fedora 19-versioon ja asennuspaketin haen osoitteesta "http://resources.ovirt.org/releases/3.3.4/rpm/Fedora/19/noarch/". Tarvitsen ainoastaan release-paketin asentamiseen. Asennan paketin komennolla "yum localinstall <paketin-osoite>". Palomuurisäädöksiä en aseta, sillä poistin sen käytöstä Fedorassa heti aluksi.

5.2.3 Valmis jakelu

oVirt tarjoaa kotisivuillaan osoitteessa "http://resources.ovirt.org/releases/3.3/iso/" oVirtin node-levykuvia, jotka ovat käytännössä Fedora-Linuxeja, joihin on asennettu oVirtin vaadittavat paketit. Node vaatii lähinnä salasanan määrittämisen. Lisäksi ovirt-engine kohdassa täytyy määrittää vielä ovirt-enginen IP-osoite ja network-asetuksista noden oma IP-osoite (kuva 20).

```

NIC Details: ens32
Driver:      e1000
Link Status: Disconnected
Vendor:      Intel Corporation
MAC Address: 00:50:56:9e:2f:c4

IPv4 Settings
Bootprotocol: ( ) Disabled ( ) DHCP (X) Static
IP Address:   192.168.200.16____ Netmask: 255.255.248.0____
Gateway:      192.168.200.1____

IPv6 Settings
Bootprotocol: (X) Disabled ( ) Auto ( ) DHCP ( ) Static
IP Address:   _____ Prefix Length: _____
Gateway:      _____

VLAN ID:      _____

Use Bridge:    [ ]

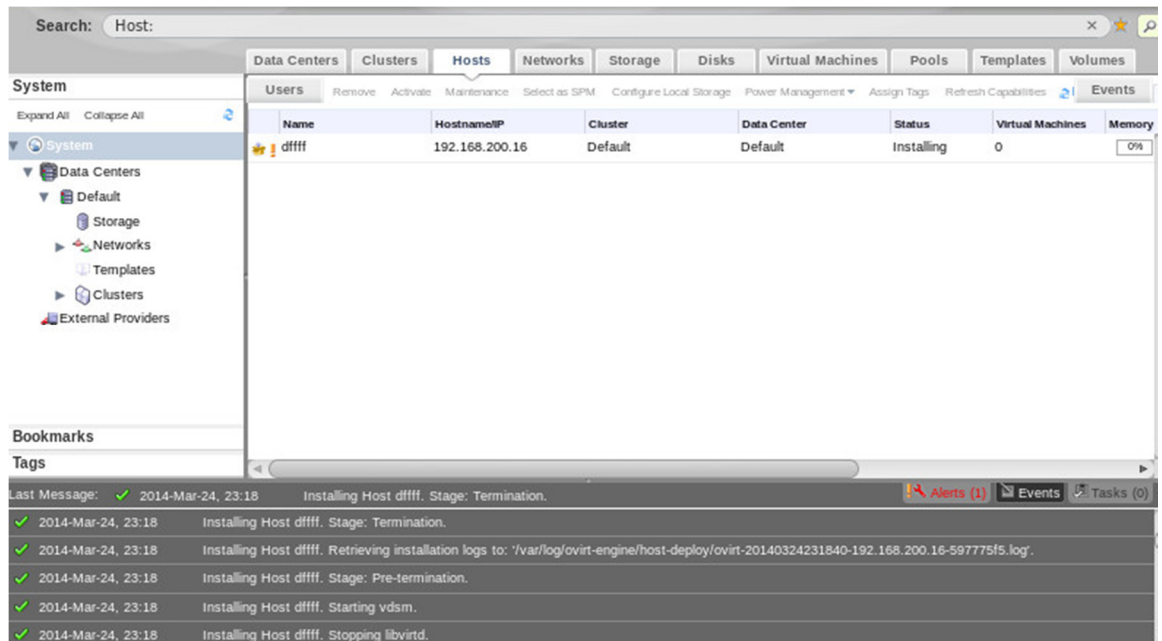
< Flash Lights to Identify >
< Save >      < Close >

```

Kuva 20. oVirt node konfiguroituna toimintakuntoon.

5.3 oVirt noden lisääminen

Ovirt-Node lisätään Administrator Portalin kautta menemällä host-välilehdelle. Ympäristössäni node oli asennettu Fedoraan, eikä se automaattisesti yrittänyt yhdistää ovirt-enginelle, joten käytin yhdistämiseen SSH-yhteystapaa. Kun asensin noden suoraan oVirtin tarjoamasta node-levykuvasta, niin node yritti yhdistää enginelle IP-osoitteen määrittämisen jälkeen (kuva 21). Tässä tapauksessa node näkyy jo host-listassa ja pelkkä approve-napin painallus sekä salasanan antaminen riittää.



Kuva 21. Näin sen asennuksen pitäisi mennä.

Käytännössä törmäsin virhe-ilmoitukseen, jossa kerrottiin SSH-yhteyden epäonnistuvan noden ja engineen välillä. CentOS:lle asennettu oVirt menee suotuisammin engineen, mutta virheilmoituksen antaa sekin. Tässä tapauksessa virhe tuleekin vain siitä, että fencing laite puuttuu ja virranhallinta suunnitelmaa ei voi tehdä. Ilman fencing laitetta high availability -mahdollisuutta ei ole. Sama pätee myös muihin hallintaohjelmistoihin.

Mikäli kaikki meni, kuten oli tarkoitettu, niin node aloittaa pakettien lataamisen heti, kun ovirt-engine saa siihen yhteyden. Käytännössä nodea voidaan sitten hallita web-käyttöliittymästä.

Käytännössä oVirtin Fedora-version kanssa ongelmaksi ilmeni noden ja engineen välinen yhteys. engine löysi muutaman kerran noden, mutta se ei pysty siitä huolimatta niitä hallitsemaan. Virheilmoitus antaa syyksi SSH-yhteyden ongelman joko käyttäjänimessä, salasanassa tai salausavaimessa. Engine kuitenkin löytää noden, joka viittaa siihen, että jokin kommunikointi engineen ja noden välillä ajoittain on. Koska en löytänyt ratkaisua yhteisön kautta tai muualtakaan, päätin kokeilla CentOS-versiota oVirtistä.

CentOS-versio toimii paremmin, sillä engine löytää noden helposti. Silti noden lisääminen ei onnistu, koska engine antaa virheilmoituksen "Install failed". Yritin etsiä ratkaisuja yhteisön kautta, mutta yksiselitteistä ratkaisua en onnistunut löytämään. Keskusteluista löytyi palomuurista, mutta se ei päde tähän tapaukseen, koska palomuuuri ei ollut koskaan päällä. Toinen eh-

dotus oli version päivittäminen. Käytännössä kokeilin kahta eri versiota (3.1 ja 3.3), mutta siitä huolimatta node ei toimi. Varsinkin valmiiksi kootun levykuvan kohdalla tilanne on outo, sillä se on tarkoitettu vain ja ainoastaan liitettäväksi engineen. Käytännössä tuo ominaisuus ei kuitenkaan toimi. Engine vaikuttaa käyttävän SSH-yhteyttä nodelle, mutta jostain syystä virheilmoitus tulee liittymisvaiheessa.

Toinen tapa käyttää ovirtia on asentaa node suoraan engineen. Tämäkään tapa ei tuottanut tulosta, sillä setup ei suostunut asentumaan loppuun saakka, vaan jäi herjaamaan “komentoa ei voitu suorittaa”.

Käytin ovirt.org-sivuston Mail-list-palvelua avuksi ongelmaa ratkoessani. Sivustolta löytyi muitakin käyttäjiä, joilla oli sama ongelma. Käyttäjä nimeltä Sven Kieske kertoo samankaltaisesta ongelmasta. Hänelläkään ei noden asennus toiminut ja logista löytyy samainen virheraportti joka viittaa SSH-yhteyteen. Kieske saa oman nodensa toimimaan erään toisen käyttäjän linkittämällä oVirt-levykuvalla, joka ei omassa ympäristössäni käynnistynyt. Levykuva jäättyi kesken asennuksen lataamisen. (Mail Archive.com 2013)

Lopulta ratkaisin ongelman kokeilemalla uutta 3.3.4-versiota oVirtistä, kun sellainen tuli saataville. Sen avulla pääsin ongelmasta eroon.

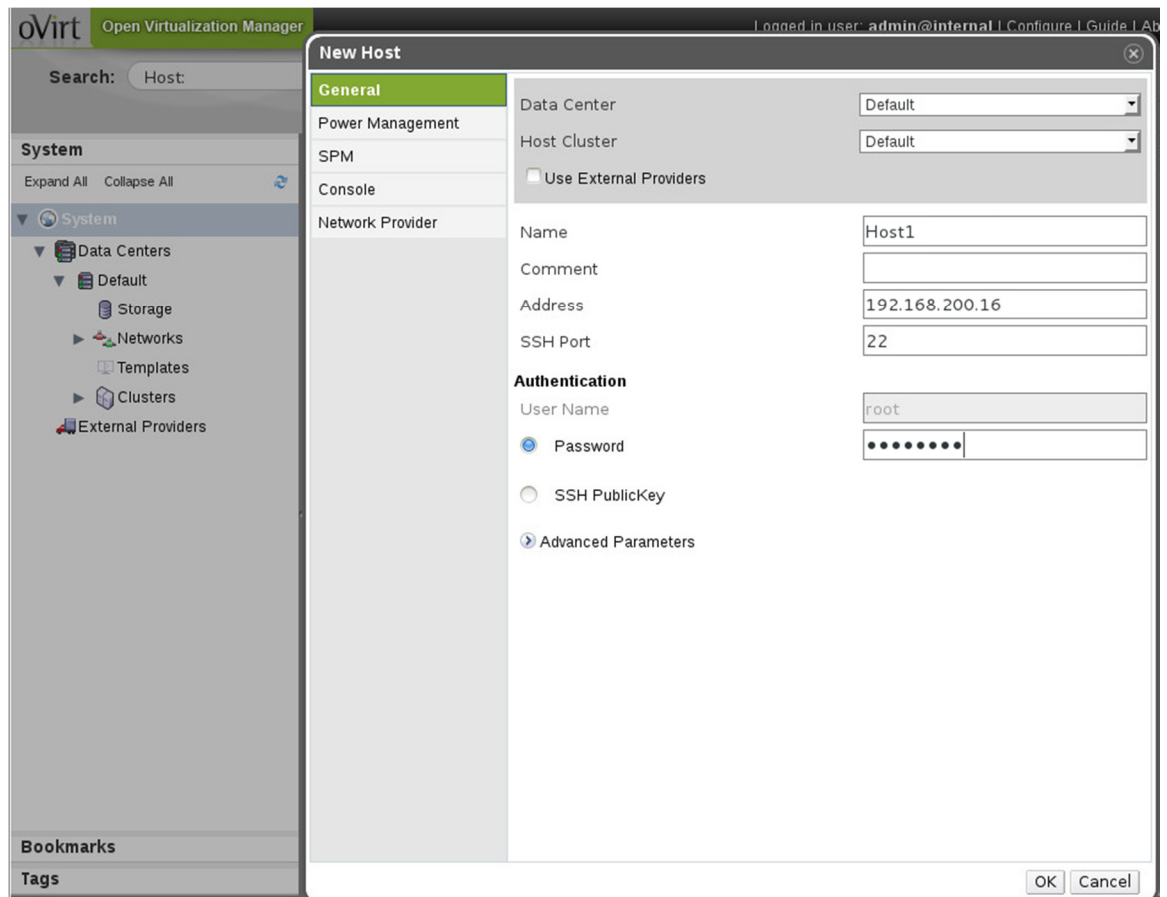
5.4 oVirtin työkalut

Konfigurointi aloitetaan osoitteesta "http://<Ovirtin IP-osoite>:80", joka on Ovirtin hallintapaneeli. Hallintapaneelistä löytyy kaksi erilaista portaalia. Toinen niistä on suunnattu käyttäjille ja toinen ylläpitäjille.

Tärkeimmät logit, testieni aikana, löytyvät ovirt-enginen "/var/log/ovirt-engine/" polusta. Siellä on logi tiedostoja eri komponenteille tai ominaisuuksille.

5.5 Administration Portal

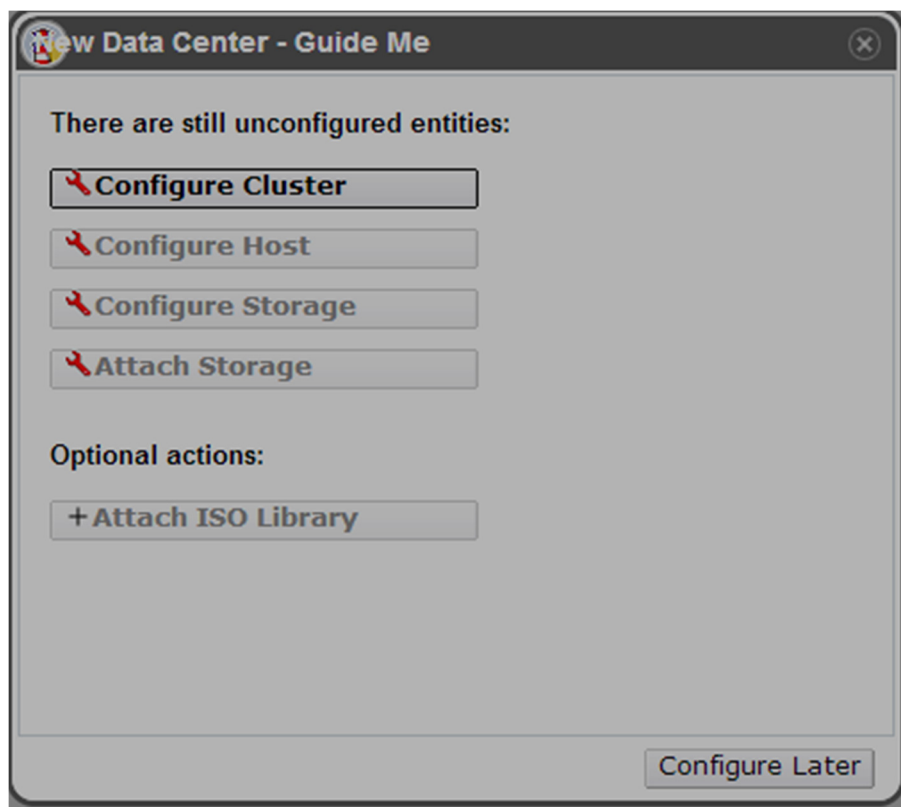
Administration Portal eli AP sisältää tiedot ylläpitäjälle tärkeistä kohteista. AP:ssa määritetään ja monitoroidaan klusteria. Käytännössä AP:n kautta lisätään uusia Hosteja, Datastoreja, verkkoja ja templateja (kuva 22).



Kuva 22. Administration Portalista voi lisätä vaikkapa uuden hostin.

5.5.1 Guide Me

Data Centers -valikossa ylläpitäjä voi määrittää ympäristön opastetusti. Ensimmäiseksi luodaan klusterin asetukset (kuva 23).



Kuva 23. Opastettu datacenterin luomiseen tarkoitettu ominaisuus helpottaa valikoiden löytämistä.

Klusterille määritetään nimi ja CPU-tiedot. Klusterille voi asettaa virransäästö sääntöjä ja muutamia optimointeja. Esimerkiksi prosessorin säikeet voidaan laskea ytiminä. Memory balloon -ominaisuudella oVirt-isännän muistia voidaan käyttää dynaamisesti virtuaalikoneille, jolloin muistia voidaan hyödyntää yli määrätyn. Myös virtuaalikoneiden migraatio-ominaisuutta voidaan rajoittaa klusterin luomisen aikana, jos sille on tarvetta.

Toinen vaihe Data Center -ominaisuudessa on oVirt-isännän liittäminen. Kuten jo aiemmin käsittelin, niin uusi oVirt-Hosti lisätään laittamalla osoitteen ja salasanan tai osoitteen ja julkisen SSH-avaimen kenttään. Tarvittaessa virranhallinta-asetuksia on mahdollista muokata. Se on suositeltavaakin, sillä muutoin siitä tulee herjaa myöhemmin.

Kolmas vaihe on storage. Oletuksena oVirt tarjoaa NFS- ja iSCSI-storagea. Discover Targets -kenttään voi syöttää kohteen IP-osoitteen, jolloin storagen pitäisi ilmestyä listaan. Valmiiksi konfiguroidun domainin voi lisätä Import Domain-valikosta, jolloin kenttään voi syöttää suoraan NFS- tai iSCSI-polun. Myös paikallinen asema käy storageksi.

Viimeisessä vaiheessa määritetään virtuaalilevyjä. Levyille määritetään koko ja muutamia ominaisuuksia. Ominaisuuksiin kuuluu myös Thin Provisioning, jolloin levyille määritetään maksimi koko storagessa, mutta levy itsessään käyttää vain tarvittavan määrän tilaa ja kasvaa siitä maksimiin tarpeen mukaan. Myös erillinen Lun-asema on tuettuna iSCSI:sta.

5.5.2 Käyttäjät

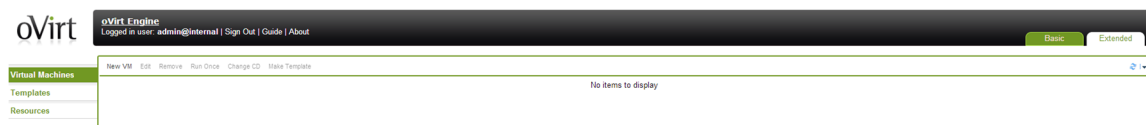
Käyttäjät perustuvat Unix-käyttäjiin. oVirt ei sisällä muuta käyttäjienhallintaa ilman kolmannen osapuolen ohjelmistoja. Siitä löytyy kuitenkin tuki LDAPille OpenLDAPin kautta ja Active Directory -tuki Microsoftin tuotteelle.

5.5.3 External Providers

External Providers -ominaisuus tarjoaa levykuvia kolmannen osapuolen palveluilta. Foreman- ja OpenStack-tyyppiset levykuvat ovat tällä hetkellä tuettuina.

5.6 User Portal

User Portal eli käyttäjäportaali on suunnattu virtuaalikoneiden käyttäjille. Sieltä käyttäjä hallinnoi ja käyttää virtuaalikoneita ja templateja. User Portalissa on keskitytty yksinkertaisuuteen, eikä käyttäjälle anneta kuin käyttöliittymä virtuaalikoneiden tekemiseen (kuva 24).



Kuva 24. oVirtin User Portalin käyttöliittymä on hyvin yksinkertaistettu.

5.6.1 Virtuaalikoneen luominen

Virtuaalikoneen tekeminen tapahtuu valmiista templatesta. Virtuaalikoneen käyttämiä resursseja voidaan vielä muokata luomisen jälkeen.

Advanced Options -valikko mahdollistaa joustavamman virtuaalikoneen luomisen. Virtuaalinen CD-asema on tuettuna ja thin provisionin on valittavissa. USB ja virtuaalinen äänikortti on mahdollista lisätä, kuten myös lisää levyasemia. Spice on mahdollista lisätä virtuaalikoneen luomisen aikana uuteen virtuaalikoneeseen.

5.6.2 Template

Templatea tehtäessä määritetään muutamia resursseja virtuaalikoneelle (kuva 25). Templatea luodessa templatelle annetaan nimi, resurssit, prioriteetti, käyttöluvat, käyttöteknologia ja levykuva. Käyttäjät voidaan ohjata käyttämään templatesta käännettyjä virtuaalikoneita, jolloin virtuaalikoneet ovat aina samanlaisia. Templateja voi tehdä lisää valmiista virtuaalikoneista.

General Network Interfaces Disks Events Permissions					
Name:	Blank		Defined Memory:	1024 MB	
Description:	Blank template		Number of CPU Cores:	1 (1 Socket(s), 1 Core(s) per Socket)	
Host Cluster:	Default		Number of Monitors:	1	
Operating System:	Other OS		Highly Available:	No	
Default Display Type:	SPICE		Priority:	Low	
			USB Policy:	Disabled	
			Origin:	RHEV	
			Is Stateless:	false	

Kuva 25. Templatelle määritetyt asetukset nähdään yksityiskohtaisesti.

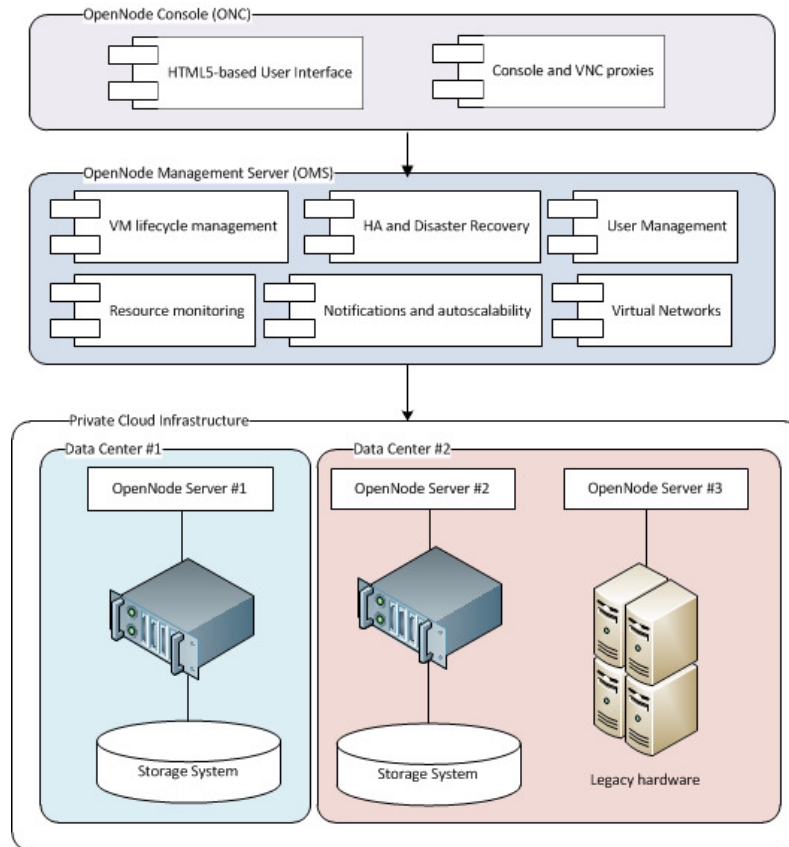
6 OPENNODE 6

OpenNode on avoimen lähdekoodin projekti, joka on suunnattu julkiselle sektorille, yksityiselle sektorille ja HPC-keskuksille. Projektin kanssa yhteistyössä ovat Saltstack, Turnkey Linux, CentOS ja Twisted.(OpenNode a 2014), (OpenNode b 2014)

OpenNode kertoo vahvuuksiensa olevan avoimen laitteiston, lisenssimaksuttomuuden, helppokäyttöisyyden ja monipuolisuuden puolella. (OpenNode c 2014)

6.1 Rakenne

OpenNode rakentuu front-endistä ja nodeista. Proxmoxin tapaan kyseessä on bare-metal-ratkaisu, jolloin kaikki tarvittava tulee samassa asennuspaketissa/levykuvassa. Käytännössä pelkkä front-end-asennus TUI-käyttöliittymällä riittää virtuaalikoneen käyttöön ja luomiseen. Web-palvelu on ainoa erillinen palanen, jonka asentaminen ei ole pakollista (kuva 26). Sen asentamiseksi OpenVZ-virtuaalikone täytyy ladata ja suorittaa.



Kuva 26. OpenNoden rakenne. (OpenNode d 2014)

6.2 OpenNode OS 6:n asennus

OpenNode vaatii keskusmuistia 4-8 gigatavua, riippuen rakennettavasta ympäristöstä. Käytän OpenNode OS 6 Update 2-versiota. (OpenNode e 2014)

OpenNode asennetaan levykuvasta, joka on saatavilla OpenNoden kotisivuilta. Suoritan asennuksen Deployment-guiden mukaisesti. Deployment-guide on saatavilla osoitteesta: "<https://support.opennodecloud.com/wiki/doku.php?id=start>". OpenNoden asennusprosessi ei eroa normaalin CentOSin asennuksesta. Käytännössä määritän vain samat asiat kuin yleensäkin käyttöjärjestelmän asennuksessa. Varsinainen konfigurointi tapahtuu käyttöjärjestelmän asennuksen jälkeen.

Kun sain OpenNoden asennettua, niin kirjauduin sisään root-käyttäjällä. OpenNoden front-end kehottaa käyttämään opennode-komentoa konfigurointiin, joten niin myös teen. OpenNodea voi käyttää jo sellaisenaan virtuaalikoneiden luomiseen, mutta itse asennan OMS:n eli

OpenNode Management Serverin. Se on käytännössä web-käyttöliittymä, joka on tässä vaiheessa vasta beta asteella.

OpenNoden web-hallintatyökalu toimii niin, että ladataan OMS:n OpenVZ-container ja asennetaan se OpenNoden sisälle. OpenNoden OMS-valikossa on painike lataamiselle, luomiselle ja rekisteröinnille. Latauspainike lataa automaattisesti levykuvan ilman sen suurempia valintoja. Näyttööni tulee ilmoitus latauksen onnistumisesta. Latauksen jälkeen asennan OMS levykuvan. Asennus kysyy IP-osoitteen ja resurssit, jonka jälkeen OMS on valmis (kuva 27). Asennuksen jälkeen se rekisteröidään OpenNodessa.

```

USwap min/max:      0 / 3.75
Number of CPUs:      1
CPU number min/max:  1 / 1
CPU usage limit (%): 50
CPU usage min/max:   50 / 100
Disk size (GB):      8.516
Disk size min/max:   2 / 8.516

IO Priority:          ( ) Low
                    (* ) Default
                    ( ) High

Bind mounts:          /src1,/dst1:/srcN,...

Hostname:             Openmanagement
IP-address:           192.168.200.17
Nameserver:           172.31.14.130

Root password:        *****
Root password x2:     *****

OS Template:          redhat

[*] Start VM          [ ] Start on boot

Create VM             Back
  
```

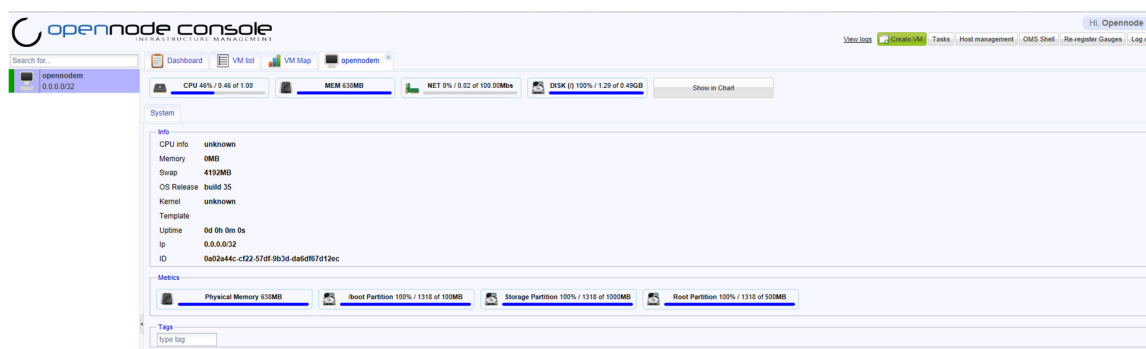
Kuva 27. OMS-virtuaalikone luodaan templatesta.

Ensimmäinen ongelmatilanne tuli, kun rekisteröinti ei toiminut. OMC host -lista näyttää tyhjältä, vaikka rekisteröintiä yritän hostin kautta. Pienen etsimisen jälkeen löysin ratkaisun virallisilta foorumeilta. Eräs ketjuun vastaaja kehottaa enableimaan opennode-dev-repositorion ja päivittämään paketit. Siispä kirjoitan komennon "yum update --enablerepo=opennode-dev opennode-tui" konsoliin ja kokeilen rekisteröintiä uudestaan. Tällä kertaa hosti löytyy OMC listasta ja voin lisätä sen klusteriin. Toimivan OpenVZ-hostin jälkeen lisään fyysisen Open-

Node hostin, jonka asennan samasta levykuvasta. Asennuksen edetessä samat ongelmat ilmenevät rekisteröimisessä, mutta edellinen ratkaisu toimi tälläkin kertaa.

6.3 Beta-asteella

OMS on tällä hetkellä beta-asteella, joka tarkoittaa sitä että siitä löytyy hyvin paljon ongelmia. Ensimmäinen ongelma tulee siitä, että Java kaatuu selaimesta riippumatta. Testasin OMS:ää Firefoxilla, Google Chromella ja Internet Explorerilla, mutta jokainen selain kaatuu muutamman minuutin käytön jälkeen. Toisinaan hallintapaneeli antaa “502 Bad Gateway”- virheen, jolloin OMS lähtee toimimaan käynnistämällä palvelimen uudelleen. Tekijät ovat myös itse tämän tiedostaneet, ja siitä syystä “beta-leima” onkin laitettu OpenNoden sisällekin näkyviin. Kuvassa 28 esitetään OpenNoden käyttöliittymä toimintakunnossa.



Kuva 28. OpenNoden käyttöliittymä beta-asteella.

6.4 Ominaisuudet

OpenNoden ominaisuudet eivät välttämättä toimi KVM-ympäristössä. OpenVZ:n puolelta ominaisuuksia löytyy muutama enemmän.

6.4.1 Template

OpenNode on yksi niistä hallintatyökaluista, joissa luodaan ensin template. Toistaiseksi templatien luominen tapahtuu valmiiksi tehdyistä levykuvista, jotka ovat tällä hetkellä Ubun-

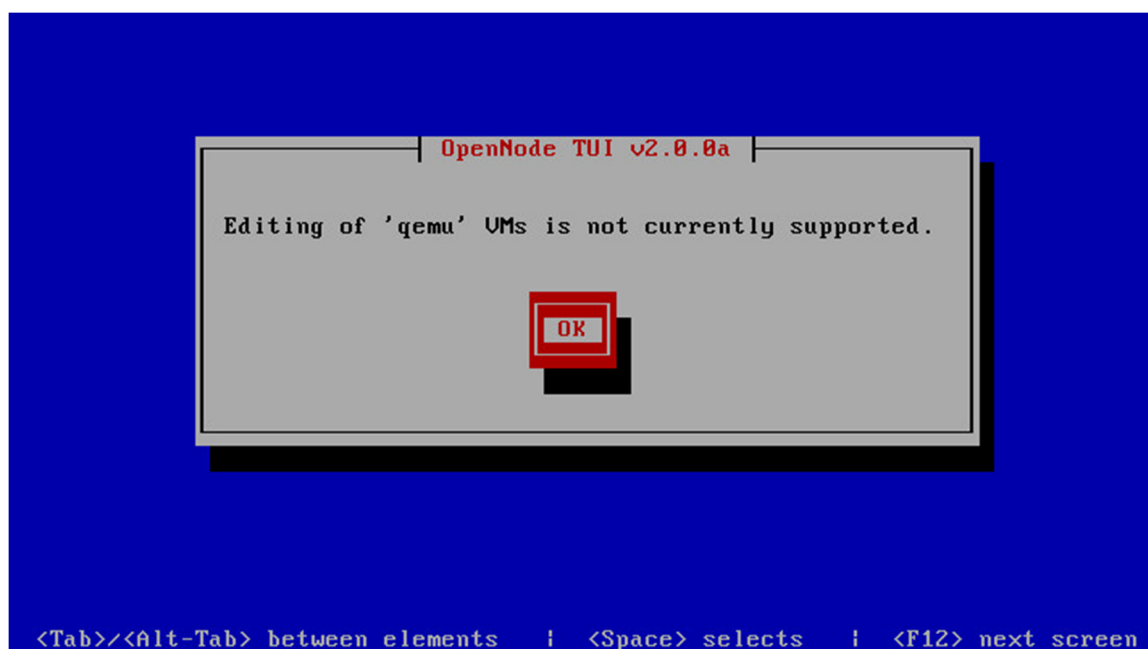
tun-palvelinversio ja CentOS. Mikäli valmiita valmiita virtuaalikoneita on siirretty ympäristöön, voi käyttäjä tehdä niistä templateja.

6.4.2 Migraatio

Migraatio ominaisuus on toiminnassa ainoastaan OpenVZ-virtuaalikoneille. Mikäli KVM-virtuaalikonetta yrittää migratoida, niin käyttäjälle ilmoitetaan ominaisuuden olevan vain OpenVZ-tuettu.

6.4.3 Virtuaalikone

OpenNode tukee vain KVM ja OpenVZ virtuaalikoneita. Käytännössä KVM-tuki on heikko, sillä suurin osa ominaisuuksista ei toimi vielä KVM-virtuaalikoneilla (kuva 29). Migraation lisäksi kloonaminen ja ominaisuuksien muokkaaminen ei ole mahdollista.



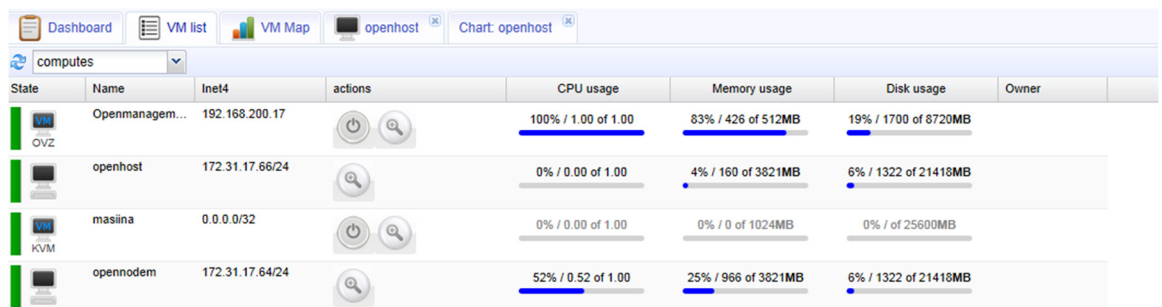
Kuva 29. OpenNoden KVM-tuki on heikko.

6.4.4 Storage

Toistaiseksi OpenNode tukee vain paikallista datastorea. Storage Poolien luominen on mahdollista. Storage Poolin luominen on mahdollista hostin storage-valikosta, jossa OpenNode kysyy vain Poolin nimen.

6.4.5 Web-UI

Web-käyttöliittymä on saatavilla OMS:n IP-osoitteesta http- ja https-versioina. Web-UI:n kautta käyttäjä voi nähdä ja liittää omia hostejaan saman näkymän alle, jolloin niiden monitorointi on helpompaa. Monitorointi näyttää kaikki tavallisimmat resurssit ja niiden käyttöasteen niin hosteista, kuin myös virtuaalikoneista. Myös hostien ja virtuaalikoneiden IP-osoitteet ovat näkyvillä, jonka lisäksi hosteista näkyvät myös verkkokortit (kuva 30).



State	Name	Inet4	actions	CPU usage	Memory usage	Disk usage	Owner
VM	Openmanagem...	192.168.200.17	[Power] [Search]	100% / 1.00 of 1.00	83% / 426 of 512MB	19% / 1700 of 8720MB	
OVZ	openhost	172.31.17.66/24	[Search]	0% / 0.00 of 1.00	4% / 160 of 3821MB	6% / 1322 of 21418MB	
VM	masiina	0.0.0.0/32	[Power] [Search]	0% / 0.00 of 1.00	0% / 0 of 1024MB	0% / 0 of 25600MB	
KVM	opennodem	172.31.17.64/24	[Search]	52% / 0.52 of 1.00	25% / 966 of 3821MB	6% / 1322 of 21418MB	

Kuva 30. Web-käyttöliittymästä näkyvät virtuaalikoneet.

Web-käyttöliittymän kautta voi myös luoda virtuaalikoneita. “Create VM”-valikosta hyppää esiin neliosainen taulukko, jossa kysytään template, host, verkkoasetukset ja root-salasana. Migratointi on myös mahdollinen VM-listasta, mutta vain OpenVZ-virtuaalikoneille.

6.5 High Availability

HA-mahdollisuudet ovat olemassa myös OpenNodessa. Käytännössä se ei tarjoa kovin paljoa KVM-puolelle, sillä manuaalinen migratoiminen tai kloonaaminen ei ole vielä tuettuna KVM-puolella. OpenVZ:lle sen sijaan on olemassa ohjeistus kotisivuilla. Sivun lopussa mai-

nitaan yksinkertaisesta High Availability -mahdollisuudesta KVM:lle, jos kehittäjiin ottaa yhteyttä.

6.6 Virtuaalikoneen luominen

Uusi virtuaalikone käännetään valmiista template-pohjasta virtuaalikoneeksi (kuva 31). Koska OpenNoden KVM-tuki on hyvin heikko, niin templatet on luotava manuaalisesti käyttämällä esimerkiksi virt-manager-ohjelmaa tai siirtämällä toisesta KVM-ympäristöstä.

Create a New Application

1. Pick a template
No templates available

2. Define VM Resources
VM Profile:
Allocation policy: **Automatic**
Memory: 0.5 Swap: 0.5
CPUs: 1 Disk: 1
Storage location: Default

3. Set network parameters
Hostname: IP Address:
Nameservers:

4. Set credentials
Root Password:
Root Password (x2):
Start VM: ☐ Start on boot: ☐

Cancel **Create**

Kuva 31. OpenNodessa virtuaalikoneen luominen alkaa valmiista templatesta kohdassa 1.

6.6.1 Templaten luominen

KVM-hostilla sijaitsevasta virtuaalikoneesta voi tehdä templatien käyttämällä OpenNoden TUI-käyttöliittymää. Kyseinen käyttöliittymä aktivoidaan front-endissä "opennode" komennolla.

Manage-valikosta löytyy Template-valikko, josta ilmestyy valikko, jossa valitaan tekniikka uudelle templatelle. KVM-valinnan jälkeen ohjelma listaa sammutettuja virtuaalikoneita, jotka sijaitsevat KVM-hostilla. Niistä jonkun valitsemalla virtuaalikoneen tiedot tallentuvat templatelle, jota ei toistaiseksi voi muokata juurikaan.

6.6.2 Valmis virtuaalikone

Create VM-painikkeesta painamalla ilmestyy valikko, josta löytyy OpenVZ ja KVM templatet. Niistä KVM-templatien valitsemalla saadaan uusi KVM-host käyttöön.

6.7 Roadmap

Kirjoittamishetkellä OpenNoden seuraavat merkittävät kehitysaskeleet tapahtuvat seuraavilla osa-alueilla:

- Shared storage (NFS, ISCSI mounts) management support in TUI/OMS
- Snapshots (backup) support for local storage pools
- Ploop OpenVZ VE disk image support
- Openvswitch support
- VLAN management support in OMS
- Firewall inclusion and management support in TUI/OMS
- KVM Spice support

- OpenVZ VE with networking support in TUI/OMS

OpenNoden roadmap löytyy virallisilta kotisivuilta ja näyttää testien mukaan olevan ajan tasalla. (OpenNode f 2014)

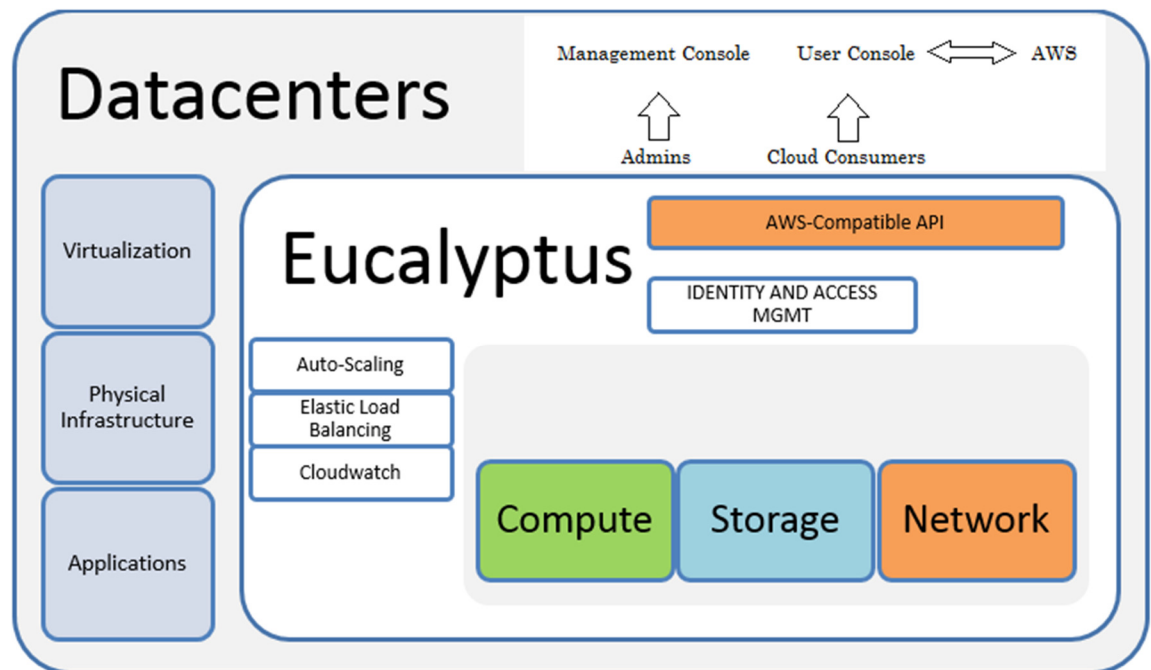
7 EUCALYPTUS 3.4.2

Eucalyptus on avoimen lähdekoodin projekti, joka tarjoaa ratkaisuja hybrid-pilviympäristöön. Projektin yhteistyökumppanina toimii Dell. Eucalyptus oli aikoinaan Kalifornian yliopiston tietotekniikkapuolen tutkimusprojekti. (Eucalyptus a 2014)

7.1 Rakenne

Eucalyptus-ympäristö rakentuu useista eri osista, jotka jatkavat datacenterin palvelut aina pilveen saakka. Eucalyptuksen komponentteja ovat (kuva 32):

- Cloud Controller
- Cluster Controller
- Node Controller
- Storage Controller
- VMware Broker
- Walrus



Kuva 32. Eucalyptuksen komponentit rakentavat tällaisen kokonaisuuden.

7.1.1 Cloud Controller

Eucalyptuksen Cloud Controller monitoroi saatavilla olevia resursseja useilta komponenteilta. Sen tarkoituksena on antaa tietoa Cluster Controllerille siitä, kuinka ympäristön resurssit ovat jakautuneet. (Cloudbook.net a 2014)

7.1.2 Cluster Controller

Eucalyptuksen Cluster Controller luo instansseja Cloud Controllerin tietojen perusteella ja määrää sen, mille nodelle uusi instanssi menee. Se hallitsee myös virtuaalisia verkkoja ja kerää tietoja nodeista, joita se sitten raportoi Cloud Controllerille. (Cloudbook.net b 2014)

7.1.3 Node Controller

Eucalyptus-nc eli Node Controller on käytännössä KVM-hypervisor. Lisätyt node controllerit näkyvät käyttöliittymässä vain klusterina. Olemassa olevat yksittäiset nodet voidaan listata komennolla “euca_conf --list-nodes” (kuva 33).

```
[root@euca eucalyptus]# euca_conf --list-nodes
NODE    CLUSTER01          192.168.200.15  ENABLED
INSTANCE    i-B9EC45CA
[root@euca eucalyptus]#
```

Kuva 33. Kuvassa näkyvät olemassa olevat Eucalyptuksen nodet.

7.1.4 Storage Controller

Eucalyptuksen Storage Controller on ominaisuus, jolla hallitaan datastoreja. Sen käyttöön ei ole graafista työkalua, vaan kaikki konfigurointi tapahtuu komentoriviltä. (Eucalyptus c 2014)

Storagen määrittämiseen voidaan käyttää euca-modify-property-työkalua. Työkalu löytyy siltä hostilta, jonne se asennusvaiheessa on asennettu. Käytännössä työkalulla voidaan valita esimerkiksi paikallinen osio, joka annetaan pilven käyttöön. Työkalua käytetään tällöin komennolla "euca-modify-property -P <OSIO>.storage.blockstoragemanager=overlay". Käytännössä erilaisille tallennusratkaisuille käytetään erilaista blockstoragemanageria, riippuen tallennustilan tyypistä. Käytössä olevat osiot näkyvät komennolla "euca-describe-services -E".

7.1.5 VMware Broker

VMware Broker on vain subscriptionin omistajille suunnattu komponentti, joka mahdollista Eucalyptuksen kommunikoinnin VMwaren infrastruktuurin komponenttien kanssa. Komponentteja ovat esimerkiksi ESXI ja vCenter. VMware Broker on yhteydessä Cluster Controlleriin ja vCenteriin. (Eucalyptus d 2014)

7.1.6 Walrus

Walrus mahdollistaa virtuaalikoneiden luomisen, poistamisen ja muun hallinnoinnin Amazon Web Servicesin Simple Storage Service -palvelua käytettäessä. Käytännössä se tallentaa käyttäjän objekteja ”/var/lib/eucalyptus/bukkits”-kansioon, josta niitä voi käyttää.

7.2 Asentaminen

Eucalyptuksesta on ainakin kaksi erilaista asennusvaihtoehtoa. Automatisoitu asennusohjelma on tarkoitettu testaukseen. Se on valmiiksi konfiguroitu sisältäen KVM-hypervisorin ja CentOS-käyttöjärjestelmän. Asennuksen luvataan olevan valmis puolessa tunnissa. Mukautettu asennus on suunnattu kokeneemmille käyttäjille. Siinä käyttäjä asentaa haluamaansa käyttöjärjestelmään Eucalyptuksen paketit itse. Lisäksi Community Cloudia voi testata Eucalyptuksen tarjoamilla palvelimilla. (Eucalyptus c 2014)

7.2.1 Mukautettu asennus CentOSille

Ensimmäiseksi lisää riippuvuudet uuteen CentOS-asennukseen käyttäen virallista Eucalyptuksen asennusohjetta osoitteessa ["https://www.eucalyptus.com/docs/eucalyptus/3.4/index.html#install-guide/eucalyptus.html"](https://www.eucalyptus.com/docs/eucalyptus/3.4/index.html#install-guide/eucalyptus.html). Verkkoyhteyden asetukset jätän myöhempään konfigurointiin, sillä joudun myöhemmin vaihtamaan IP-osoitteen omaan ympäristööni sopivaksi.

SELinuxin konfiguroin /etc/selinux/config-tiedostoon arvolla SELINUX=permissive, sillä Eucalyptus tarvitsee kyseiset arvot sallituksi. Muutokset hyväksyn komennolla “setenforce 0”.

Loput tarvittavista riippuvuuksista ovat NTP-daemon ja Palomuuuri. Tällä hetkellä NTP on asennettu ja arvot ovat jo oletuksena asetettuna oikein. Myös palomuuuri on poissa päältä tässä CentOS asennuksessa.

Seuraavaksi asennan CentOSille EPEL-repositoriosta Eucalyptuksen asennuspaketit `eucalyptus-release-3.4.noarch.rpm`, `eucalyptus-cc-release-3.0.noarch.rpm`, `eucalyptus-sc-release-3.0.noarch.rpm` ja `eucalyptus-walrus-release-3.0.noarch.rpm`. Muutamia lisäosia joudun jättämään pois, sillä en ole Eucalyptuksen tilaaja. Niiden joukossa ovat VMware Broker ja SAN.

Jatkan asennusta Cloud Controllerin asennuksella ”`yum install eucalyptus-cloud`” -komentoa käyttäen. Loput asennuspaketit (`eucalyptus-cc`, `eucalyptus-sc` ja `eucalyptus-walrus`) suositellaan asennettavaksi eri palvelimille, mutta ne on mahdollista asentaa myös samalle. Omassa asennuksessa käytän samaa palvelinta kaikille. Eucalyptus load balancer on myös tarjolla, mutta en usko sillä testivaiheessa olevan käyttöä, sillä virtuaalikoneita ei tule olemaan niin monta, että balansoija tulisi käyttöön. Kaikki konfigurointi täytyy tehdä asennuksen jälkeen. Asentamiseen meni aikaa noin puoli tuntia.

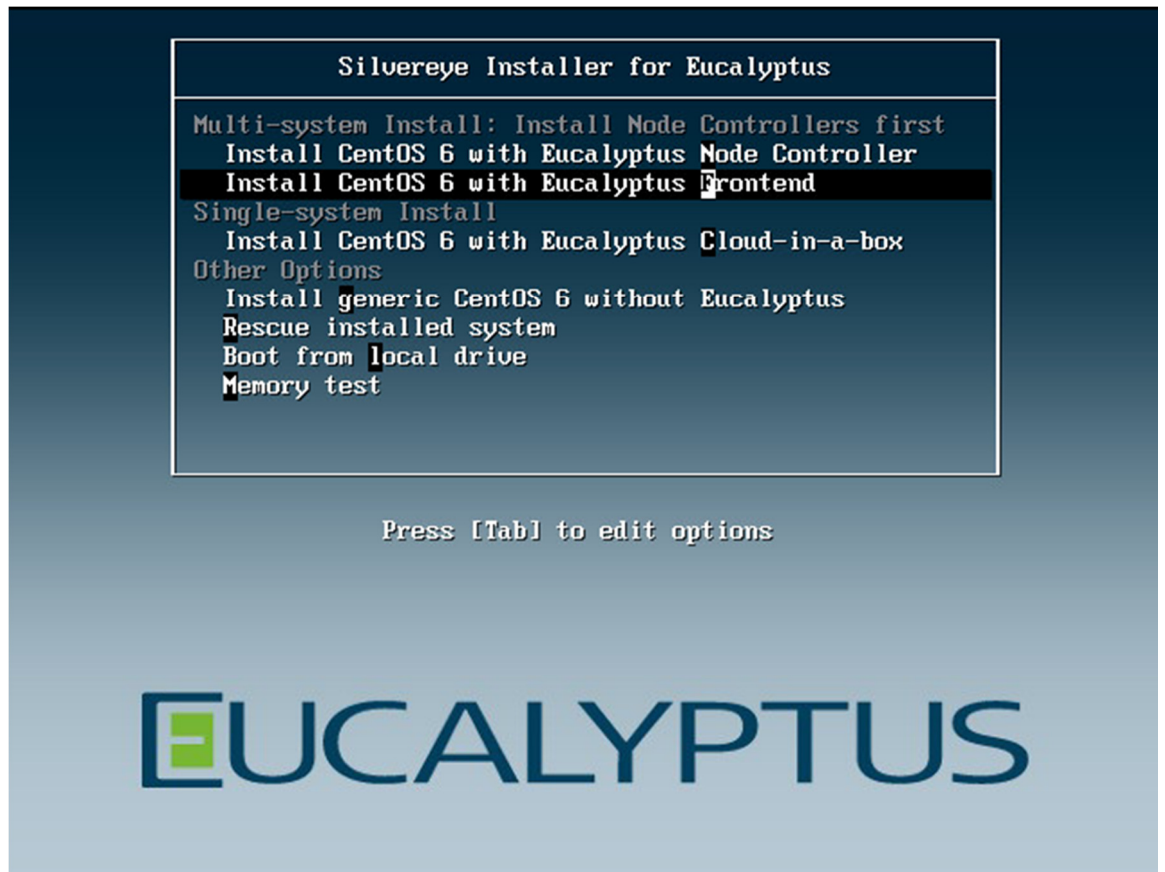
Eucalyptuksen konfiguraatio tiedosto löytyy polusta `/etc/eucalyptus/eucalyptus.conf`. Ennen kuin Eucalyptus lähtee käyntiin, se lukee kyseisen `conf`-tiedoston.

Palvelimelle tulee seuraavaksi määrittää kiinteä IP-osoite, jotta siihen pääsee käsiksi aina samasta osoitteesta. Tässä tapauksessa IP-osoitteeksi tulee `192.168.200.15` ja DNS-osoitteeksi `192.168.200.10`. Seuraavaksi määritän pilven asetukset lataamalla `euca_conf`-tiedoston komennolla ”`euca_conf --initialize`” ja käynnistän `eucalyptus-cloud`-palvelun. Ennen määrittystä on huomattava, että palvelu ei saa olla päällä.

Määrittelyn jälkeen käynnistän palvelun komennolla ”`service eucalyptus-cloud start`”.

7.2.2 FastStart

FastStart-asennus tarjoaa mahdollisuuden Eucalyptus-ympäristön asennuksen kolmella tavalla (kuva 34). Ensimmäisessä vaihtoehdossa front-end ja node asennetaan erilleen toisistaan. Silloin käytössä tulee olla kummallekin oma palvelin. Toinen vaihtoehto on asentaa Cloud-In-A-Box. Silloin front-end ja node tulevat samaan palvelimeen. Kolmas vaihtoehto on asentaa mukautetun asennuksen tavoin pelkkä CentOS-käyttöjärjestelmä ja kasata ympäristö itse Eucalyptuksen paketeista.



Kuva 34. Eucalyptuksen valmis asennusmediakin tarjoaa vaihtoehtoja samasta levykuvasta.

Jatkaakseni asennusta valitsen vaihtoehdon yksi, jossa asennan Front-Endin ja Nodet erilleen. Asennus kysyy IP-asetukset, joiden tulee pysyä samana loppuun asti. Vaihtoehtoina ovat DHCP:n kautta IP-tietojen hakeminen tai kiinteä IP-osoite. Asennuksessani käytän kiinteää IP-osoitetta 192.168.200.17.

Asennuksen edetessä Eucalyptus kysyy IP-aluetta, jolla toimia. Omassa ympäristössäni alue on 192.168.200.10-192.168.200.14, perustuen käytettävissä oleviin IP-osoitteisiini. Myös DNS-palvelimen osoite on määritettävä viimeistään tässä vaiheessa. Lisään siihen ympäristössä sijaitsevan DNS-palvelimen osoitteen.

Asennuksen jälkeen kestää hetken, ennen kuin Eucalyptus määrittää pilven ja käynnistyy.

7.2.3 Eucalyptus node

Eucalyptus noden asentaminen onnistuu FastStart-imagen kautta tai asentamalla tarvittavat paketit CentOS-asennukseen. Päättän kuitenkin käyttää FastStart-levy kuvaa ja asentamalla sitä kautta Noden.

Käytännössä noden asentaminen näyttää samalta kuin front-endin asentaminen. Määritän IP-osoitteeksi 192.168.200.15 ja saman DNS-osoitteen kuin aiemmin Front-Endille.

Uusi node controller lisätään ympäristöön komennolla “euca_conf --register-nodes [IP-osoite]”. Komennon jälkeen node controller on valmis käytettäväksi. Komennot ajetaan Eucalyptuksen front-endissä.

7.3 Asennuksen testaaminen

Asennusta voi testata asentamalla network-tomography-paketin. Paketti löytyy osoitteesta: http://downloads.eucalyptus.com/software/tools/centos/6/x86_64/network-tomography-1.0.0-3.el6.x86_64.rpm. Paketin asennuttua, testin voi suorittaa “/usr/bin/network-tomography [Cloud controller IP] [Storage Controller IP] [Cluster Controller IP] [Walrus IP] [VM Broker IP]” -komennolla. IP-osoitteet ja laitteet tulevat sen mukaan, millainen ympäristö itsellä on käytössä. Itselläni käytössä on juuri nyt vain yksi palvelin jossa on asennettuna edellä mainitussa ohjeessa asennetut palaset, joten voin testata ympäristöäni vain localhost-osoitteeseen. Testi antaa palautteeksi paljon erilaista dataa.

7.4 CloudWatch

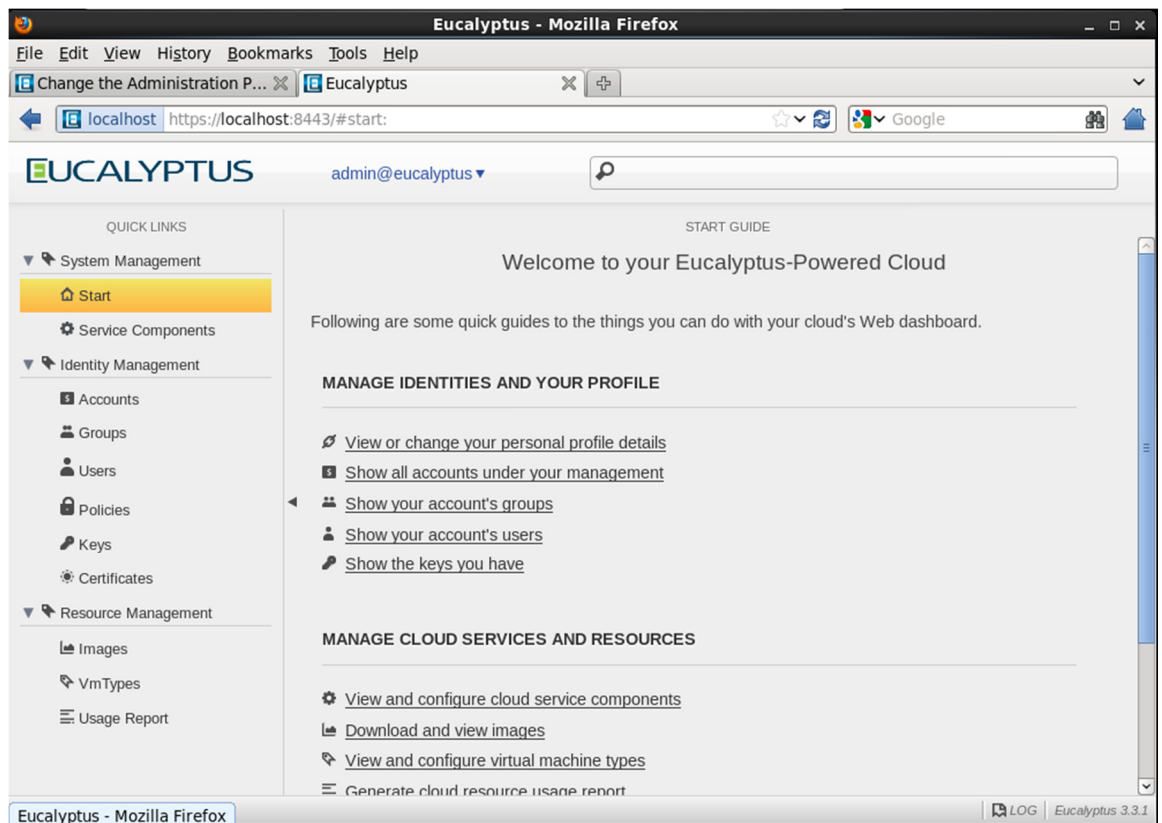
CloudWatch on työkalu, jolla monitoroidaan Eucalyptus-ympäristöä. CloudWatch varoittaa, mikäli ympäristössä tapahtuu asioita, joista ylläpitäjä haluaa tiedon. CloudWatch on hyvin konfiguroitavissa tai vaihtoehtoisesti ylläpitäjä voi käyttää oletusasetuksia, jotka ovat jo valmiina Eucalyptuksen mukana konfiguroituna. (Eucalyptus e 2014)

7.5 Ylläpito-paneeli

Eucalyptuksen ylläpito-paneeli löytyy oletuksena osoitteesta ”https://[ip]:8443” (kuva 35). Siellä on mahdollista luoda käyttäjiä ja ryhmiä sekä sääntöjä. Osoitteesta ”https://[ip]:8773” löytyy cloud control käyttöliittymä Eucalyptukselle sekä Amazon Web Servicelle.

Eucalyptuksen ylläpito-paneeli on tarkoitettu profiilien, palveluiden sekä resurssien hallintaan ja monitorointiin. Järjestelmään liittyvät komponentit on listattu Service Componentsin alle, josta ylläpitäjä näkee palveluiden osoitteet, tyypit ja portit.

Käyttäjien hallintaan Eucalyptuksesta löytyy Identity Management. Identity Managementin kautta voidaan luoda uusia käyttäjiä, ryhmiä, sääntöjä sekä sertifikaatteja. Uusi käyttäjä voi kirjautua Käyttäjä-paneeliin ja käyttää virtuaalikoneita asetettujen sääntöjen rajoissa.



Kuva 35. Ylläpitäjän paneelissa on ylläpitoon liittyviä asioita.

Eucalyptuksen levykuvat ovat valmiiksi tehtyjä samaan tapaan kuin esimerkiksi OpenNebulassa. Levykuvat ovat saatavilla EuStoressa, jota käytetään komentoriviltä. Front-endissä ajettava komento ”eustore-describe-images” listaa ladattavissa olevat levykuvat. Resource Mana-

gement näyttää asennetut levykuvat ja niiden tiedot. Uusia levykuvia voi asentaa manuaalisesti “eustore-install-image”-komennolla. Graafista työkalua levykuvan asentamiseen tai lataamiseen ei ole. Levynkuvia on mahdollista luoda itsekin Eucalyptuksen sivuilla olevan ohjeen mukaisesti, mikäli haluamaansa levykuvaa ei ole EuStoressa saatavilla. Ohje löytyy osoitteesta "https://www.eucalyptus.com/docs/eucalyptus/3.2/ug/creating_images.html#creating_images".

VmTypes-valikosta ylläpitäjä voi hallita virtuaalikoneiden resurssien käyttöä asettamalla rajoitteita virtuaalikoneiden luomisessa olevien resurssiasetusten resurssimääriä.

7.6 Käyttäjä-paneeli

Mikäli asennuspaketissa ei tule mukana käyttäjän konsolia, niin se pitää ladata erikseen. Manuaalisessa asennuksessa konsoli asennetaan itse ja FastStart asennuksessa konsoli asentuu automaattisesti. FastStart asentaa myös load balancerin. Konsolin voi asentaa komennolla ”yum install -y eucalyptus-console”

Käyttäjä-paneeli on suunnattu käyttäjille, jotka luovat virtuaalikoneita. Paneelistä löytyy versio AWS-käyttäjille, mikäli AWS on käytössä.

Dashboard kertoo tietoja virtuaalikoneiden määrästä, IP-osoitteista ja tallennustilasta. Kyseiset asiat ovat kaikki, joihin käyttäjä pääsee käsiksi tästä paneelistä.

7.6.1 Virtuaalikoneen luominen

Ensiksi on hyvä käydä laittamassa Network & Security -välilehdellä uusi avain valmiiksi, sillä sitä tarvitaan virtuaalikoneeseen yhdistämisessä.

Instances-välilehdessä käyttäjä voi luoda oman virtuaalikoneen. Virtuaalikonetta luotaessa valitaan levykuva jota käytetään. Virtuaalikoneen resurssit määritetään valmiiksi asetetuista Instance Size -vaihtoehtoista, jotka määrittää ylläpitäjä admin-paneelistä. Security-välilehdellä liitetään aiemmin luotu avain virtuaalikoneeseen.

Uusi instanssi tulee näkyviin Instances-valikkoon ja sille voidaan laittaa jokin levy käyttöön valitsemalla virtuaalikone listasta ja valitsemalla “More actions” alta “attach volume”. Levyt luodaan Storage-välilehdessä. Levyille annetaan koko ja nimi, jonka jälkeen se on käytettävissä. Käytössä olevat levyt näkyvät listassa oranssina.

Käytettäviä IP-osoitteita löytyy Network & Security -välilehdestä, johon voidaan määrittää jokin IP-alue josta virtuaalikoneille voidaan määrittää osoite. Käytössä olevat osoitteet näkyvät myös listassa yhdistettynä johonkin tiettyyn instanssiin.

7.6.2 Virtuaalikoneen käyttäminen

Virtuaalikoneiden käyttäminen onnistuu yhdistämällä ssh:n kautta virtuaalikoneen IP-osoitteeseen. Tämä kuitenkin vaatii avainta, joka on ladattavissa käyttäjä-paneelin Security-välilehdeltä. Avaimen voi esimerkiksi liittää Putty-ohjelmaan tai lisäämällä linuxin SSH-ohjelmaan “-i”-komennon.

Eucalyptus tukee VNC-teknologiaa, jos se on määritetty levykuvaa tehtäessä. Työpöytävirtualisointia ei ole näkyvillä ohjelman käyttöliittymässä. Eucalyptus kehottaa käyttäjää yhdistämään SSH-tekniikalla virtuaalikoneeseen, kun käyttäjä painaa Connect-nappulaa virtuaalikoneen valikossa

7.7 Roadmap

Eucalyptuksen tuleva versio on numeroltaan 4.0. Sen julkaisu on aikataulutettu huhtikuulle 2014. Uusina ominaisuuksina tulevat parantamaan mm. load balanceria, käyttäjien hallinnoinnin helpottamista ja uusi ”Edge Networking Mode”. AWS-tuki parantuu uudistetulla objektien tallennustekniikalla. (Eucalyptus f 2014)

8 POHDINTA

Käytin suhteellisen tasapuolisesti aikaa kaikkien hallintatyökalujen kanssa testien aikana. Tavoitteeni oli saada kaikki ympäristöt siihen kuntoon, että niissä voi luoda virtuaalikoneita. Pienen taistelun jälkeen sain kaikki toimintakuntoon, joten tavoite täyttyi.

Suuria eroja ei testattavien välille loppujen lopuksi tullut. Kaikkien hallintatyökalujen kohdalla perustoiminnot olivat kutakuinkin samoja. Pieniestä eroista suurimmat erot olivatkin siinä, millaiseen tarkoitukseen hallintatyökalu on suunniteltu. Osa hallintatyökaluista oli ns. pilvi-ideologian pohjalta tehtyjä, jolloin käyttöliittymä oli karsittu varsin alkeellisen näköiseksi, mutta samat ominaisuudet olivat silti olevassa hieman syvemmällä pinnan alla.

Teknologioissa oli kahdenlaisia ratkaisuja. Suurin osa hallintatyökaluista hyödynsi libvirt-virtualisointirajapintaa, kun taas toiset olivat ns. bare-metal-ratkaisuja. Huolimatta teknologioiden eroista, hallintatyökalun ominaisuudet pysyivät edelleen samoina. Suurin eroavaisuus tulikin asennusprosessista, jossa bare-metal-ratkaisut olivat nopeampia. Myös niiden konfigurointi toimintakuntoon oli suurilta osin valmiina heti asennuksen jälkeen.

KVM-hallintatyökaluissa on näkyvissä edelleenkin rosoista pintaa. Kirjoitusvirheet ja bugit käyttöliittymässä ovat täysin tavallisia. Kaikki testaamani hallintatyökalut olivat web-käyttöliittymällä varustettuja, eikä millään niistä ollut erillistä asennettavaa ohjelmistoa. Toisaalta myöskään selainriippuvaisuudessa ei ollut eroja, vaan Firefox, Chrome ja Internet Explorer toimivat hallintatyökalujen kanssa moitteitta. Työkalut kykenevät suurilta osin toteuttamaan tehtävänsä, etenkin jos High Availability laitetaan toimintakuntoon.

8.1 Hybrid-cloud

Kaikki testeissä olevat hallintatyökalut pystyvät datacenter virtualisointiin jollain tasolla, mutta osa on tarkoitettu myös pilvipalveluksi. Pilvipalvelun erottaminen pelkästä datacenter virtualisoinnista on joissakin tapauksissa hankalaa ja toisinaan se on pelkkä sana, joka saa datacenter virtualisoinnin kuulostamaan mielenkiintoiselta.

Pilvipalvelun erottaa pelkästä datacenter virtualisoinnista sen yksinkertaisuus. Käytännössä kaikkea yritetään tehdä niin yksinkertaista, kuin se vain on mahdollista. Toisinaan tämä vie

vapauden tunteen tekemisestä, mutta takaa sen, että ympäristö on hallinnassa ja usein myös paremmin toiminnassa. Tämä teoriani perustuu siihen, että käyttäjän tekemien virheiden määrä on todennäköisesti pienempi, kun hänelle ei anneta vaihtoehtoa tehdä jotakin asiaa väärin. Tämä on hyödyllistä varsinkin silloin, kun käyttäjä haluaa vain sen virtuaalikoneen käyttöönsä, eikä hänen tarvitse tietää virtualisointialustasta tai tietokoneen resurssien jakamisesta yhtään mitään.

8.1.1 Eucalyptus

Eucalyptus oli ainoa, jonka web-käyttöliittymästä oli poistettu kaikki pilven alapuolella oleva, kuten esimerkiksi hostien ja storagen hallinta ja monitorointi. Niiden konfigurointi hoituu komentoriviltä. Esimerkiksi storage controllerin konfigurointi hoituu kokonaan komentoriviltä. Käyttöliittymästä ylläpitäjä valitsee sen, minkä kokoisia levyjä käyttäjä voi tehdä. Käyttäjän päässä näkyy lista ylläpitäjän määrittämistä levyistä, joista voi valita omaan virtuaalikoneeseen sopivimman.

8.1.2 OpenNebula

OpenNebula hoitaa pilvipuolen maltillisemmin. Heidän periaatteidensa mukaan on tärkeää, että virtualisointi hoidetaan hyvin myös datacenterin puolella, jotta pilvessä saadaan kaikki potentiaali käyttöön. Käytännössä se näkyy siinä, että OpenNebula antaa käytettävästi konfigurointimahdollisuuksia web-käyttöliittymänsä kautta. Ylläpitäjälle annetaan kuitenkin samanlaiset mahdollisuudet yksinkertaiseen virtuaalikoneiden luomiseen, sillä virtuaalikoneet luodaan templatien kautta, jolloin ylläpitäjä voi rajoittaa käyttäjän tekemisiä jo siinä vaiheessa luomalla templatet, joita käyttäjä joutuu käyttämään. Mikäli käyttäminen on tuolloinkin liian rajoitettua, voidaan käyttäjälle antaa avaimet templatien luomiseen, jolloin hänellä on joukko levyjä ja levykuvia käytettävissään, mutta ylläpitäjä voi silti määrätä ne mediat, joita käyttäjä voi käyttää. (OpenNebula g 2014)

8.1.3 OpenNode

OpenNode on tällä hetkellä lähes käyttökelvoton pilvipalvelu web-käyttöliittymänsä ikävien bugien takia, että sitä ei voi arvioida samalla tavalla kuin toisia. Se kuitenkin tähtää pilvipalveluksi Eucalyptuksen ja OpenNebulan tavoin. Toistaiseksi se ei kuitenkaan ole toimiva pilvipalvelu, sillä datacenterin hallinnointi on vasta aluillaan. Web-käyttöliittymästä voi kuitenkin päätellä sen, että lopputulos tulee muistuttamaan OpenNebulan kaltaista palvelua, jos se sille tasolle joskus pääsee.

Vähemmän puhuttuja ominaisuuksia OpenNodesta kuitenkin löytyy. Siihen on koodattu sisälle tuki WHMCS-maksunhallintaohjelmistolle. Maksunhallinta löytyy OMS:n puolelta, joten sen varsinaista käyttöönottoa kannattaa odottaa. (OpenNode g 2014)

8.2 Datacenter virtualisointi

Datacenter virtualisointiin tarkoitetut työkalut erottuvat hybrid-cloud-ratkaisuista sillä, että niiden ominaisuuksia ei piiloteta käyttäjältä. Yksinkertaistamisen sijaan ominaisuuksia lisätään ja käyttöliittymää yritetään samalla pitää käyttökelpoisena. Testieni kohteista ainoastaan Proxmox ja oVirt olivat niitä, joita markkinoidaan virtualisointialustan hallintatyökaluina pilvipalvelun sijaan.

8.2.1 Proxmox

Proxmox keskittyy pilvipalvelun sijaan datacenter virtualisointiin. Se on nähtävissä erityisesti ominaisuuksissa, joita käyttäjälle tarjotaan. Virtuaalikoneiden luominen on tehty helpoksi perinteisellä median asettamisella virtuaaliseen CD-asemaan pilvipuolen templatien luomisen sijaan. Käyttäjälle tarjotaan kaikki ominaisuudet suoran web-käyttöliittymän kautta samassa paneelista. Pääsynvalvonta erottaa ylläpitäjän tavallisesta virtuaalikoneen käyttäjästä.

8.2.2 oVirt

oVirt on myös datacenter virtualisointiin suuntautunut virtualisointi ratkaisu. Datacenter ominaisuudet ovat samankaltaiset kuin Proxmoxissa. Virtuaalikoneet luodaan perinteisen kaavan mukaan asettamalla CD-media virtuaaliasemaan. Käyttäjän ja ylläpitäjän erottavat tässä tapauksessa jo omat portalit. Ylläpitäjä hallitsee omalla portalillaan hosteja ja storagea, kun taas käyttäjä on suljettu omaan portaliinsa.

8.3 Käytettävyys

Hallintatyökalujen käytettävyys on yksi keskeisimmistä asioista siinä, kun datacenter ympäristöön halutaan KVM-hallintaan ohjelmisto.

8.3.1 Eucalyptus

Eucalyptus on käytettävyydeltään erikoinen. Sitä käyttäessä tuntuu, että ei oikeasti hallitse ympäristöään. Käyttöliittymässä ylläpitäjä tuntee olevansa vain VIP-asiakas, sillä oikeasti sieltä ei voi tehdä mitään. Tämä ei ole yhtään liioiteltua, sillä kaikki konfigurointi datacenter ympäristöön on tehtävä asennusvaiheessa tai siltä hostilta, jossa kyseinen palvelu on. Hallintaan käytetään komentorivi työkaluja, jotka ovat ainoa mahdollisuus esimerkiksi uuden hostin tai storagen lisäämiseen ympäristöön.

Eucalyptuksen käyttäminen komentoriviltä on kuitenkin helppoa, sillä Eucalyptuksesta on paljon dokumentaatiota tarjolla. Sen kotisivuilta löytyy aiheeseen liittyviä kirjoja, PDF-ohjeita ja tietenkin yhteisö.

Käyttäjälle tai asiakkaalle Eucalyptus on hyvin yksinkertainen ainakin teoriassa. Käytännössä virtuaalikoneen luominen ja varmuuskopiointi ovat ainoat asiat mitä hän voi tehdä. Virtuaalikoneen käyttäminen onkin sitten toinen juttu. Käytännössä sen käyttämiseen tarvitaan SSH-yhteys ainakin alkuvaiheessa. SSH-yhteys ei ole toimintakuntoinen heti Linux-virtuaalikoneen asennuksen jälkeen, vaan käyttäjä tarvitsee vielä Eucalyptuksesta SSH-avaimen. SSH-avain syötetään sovellukseen, jota käytetään SSH-yhteyden avaamiseen. Jos

virtuaalikoneeseen asennetaan Windows, niin silloin täytyy luoda Windows levykuva. Levykuva muokataan käyttökuntoon Eucalyptus Windows Integration -työkalulla, jolloin siihen pääsee käsiksi RDP-protokollaa käyttäen.

Käytännössä käyttöliittymä on hyvä, mutta virtuaalikoneen käytettävyyden kannalta erittäin huono. Käyttöliittymä ei tarjoa nopeaa pääsyä virtuaalikoneeseen, eikä se ole ylläpitäjällekään mieluisa asia. Yksinkertaisuuden taustalta löytyy monimutkaisuutta. Jos virtuaalikoneiden käyttäminen tehdään helpoksi, niin kyseessä olisi käyttäjälle aikalailla täydellinen pilvikäyttöliittymä. Ylläpitäjä puolestaan tyytyy komentoriviin.

8.3.2 OpenNebula

OpenNebula on ensivilkaisulta hankalan näköinen. Käyttöliittymä näyttää enemmän ominaisuuksia kuin Eucalyptuksen vastaava. Esimerkiksi hosteja voi hallita käyttöliittymästä.

Käyttöliittymässä käytetään paljon muuttujien nimiä kuten "NETTX", "CPUSPEED", "LABEL", "MODELNAME" tai "NETRX". Tästä syystä käyttöliittymään täytyy perehtyä tarkasti, sillä osa muuttujista voi olla tärkeä osa. Käyttöliittymässä on eroteltuna Infrastruktuuriin liittyvät asiat resursseista ja käyttäjienhallinnasta. Kaikki valikot ovat loogisissa paikoissa, mutta sivuston asettelu on ruma. OpenNebulan käyttöliittymä on muuttunut vuosien jälkeen parempaan suuntaan, joten se kannattaa pitää mielessä.

Monitorointi on hyvin esillä OpenNebulassa. Virtuaalikoneen tai vaikka itse hostin tiedot näkyvät helposti. Myös käppyröitä on tarjolla virtuaalikoneista, verkosta ja resursseista heti Sunstonen ensimmäisenä asiana.

OpenNebulan käyttöliittymä on heikko. Asettelu on keskinkertaista ja tilaa hukataan ylisuuriin valikkoihin. Pilvikäyttöliittymäksi se on aivan liian monimutkainen ja datacenter käyttöliittymäksi liian epäkäytännöllinen. Virtuaalikoneita pääsee kuitenkin käyttämään suoraan käyttöliittymästä, joka tekee asiat käyttäjälle mieluiseksi työpöytävirtualisoinnin kannalta. Uudemmissa OpenNebulan versioissa on saatavilla erilaisia näkymiä datacenterille. Cloud View, User View, VDCAdmin View ja Admin View. Ne ovat käytössä OpenNebula 4.6-versiossa. Käytännössä näkymät ovat yksinkertaistettuja tai karsittuja versioita olutuksena olevasta näkymästä. Ainoana suurempana muutoksena on Cloud View, jossa käyttäjälle annetaan Eucalyptus

lyptusta muistuttava käyttöliittymä, jossa on Create-painike uuden virtuaalikoneen luontiin. Käytännön testeihin kyseinen versio ei ehtinyt, mutta on kuitenkin hyvä tietää, että käyttöliittymää hiotaan paremmaksi.

OpenNebula on suotuisin sellaiseen ympäristöön, jossa halutaan datacenterin kontrolli pitää monipuolisena ja jossa halutaan käyttää pilveen suunnattua ohjelmistoa.

8.3.3 OpenNode

OpenNoden käyttö rajoittuu tällä hetkellä TUI-käyttöliittymään. Web-käyttöliittymä on olemassa, mutta tekniikan puolesta se on paljon muita jäljessä.

TUI-käyttöliittymä on komentorivipohjainen. Se tarjoaa muutaman valikon, joista voi käytännössä luoda KVM-virtuaalikoneen templatesta, joka on luotu libvirtillä. OpenVZ-käytössä se pystyy migratoimaan ja jopa live-migratoimaan. Myös virtuaalikoneita voi tuolloin luoda joustavammin.

Web-käyttöliittymää pystyin käyttämään muutamia minutteja kerrallaan, ennen kuin selaimessa tapahtui kaatuminen. Kyse oli luultavasti javan kaatumisesta, sillä kaikki muutkin selaimet kaatuivat samalla tavalla. Ominaisuuksia on tarjolla muutamia, mutta KVM-tuki oli niin rajoitettua, että esimerkiksi migraatiota ei ollut olemassa kuin OpenVZ:lle.

Ylläpitäjä työskentelee komentoriviltä. Se on paikka josta lisätään uudet hostit ja muut ympäristöön kuuluvat laitteet. Sieltä myös luodaan templatet KVM-käyttöön.

OpenNode ei ole KVM:lle edes mikään käyttöliittymä vielä. Kuitenkin sen avulla voi spekuloida, sillä OpenVZ tarjoaa ne ominaisuudet, joita KVM:ssä tulee joskus olemaan. Se muistuttaa käyttöliittymältään OpenNebulaa, mutta karsituilla ominaisuuksilla. Virtuaalikoneen käyttäminen on mahdollista käyttöliittymästä löytyvällä päätteellä, mutta VNC-yhteys tarvitsee toistaiseksi kolmannen osapuolen ohjelmiston.

OpenNoden käyttöliittymä on yksinkertainen, mutta se johtunee lähinnä ominaisuuksien puutteesta. Yhtään syytä en keksi OpenNoden valitsemiseen sen nykyisellä tilalla. Kehitystyöstä riippuu se, nouseeko OpenNode haastamaan muita.

8.3.4 Proxmox

Proxmoxin käyttö on pääasiassa web-käyttöliittymän hallintaa. Konfigurointeja saatetaan tehdä komentoriviltä joihinkin tiedostoihin ja hostit lisätään komentorivin kautta, mutta pääasiassa web-käyttöliittymä riittää pitkälle hallinnointiin.

Proxmoxin tiivis käyttöliittymä on perusteltu, sillä sieltä löytyy paljon ominaisuuksia. Valikot on piilotettuina välilehtiin, joka on toimiva ratkaisu tällaisessa ohjelmistossa. Näkymässä tarjotaan ylläpitäjälle sekä käyttäjälle paljon tietoa ympäristöstä. OpenNebulasta tutut käppyrät löytyvät välilehtien takaa kuten myös High Availabilityn konfigurointi-tiedostot. Myös päivitykset tarjotaan käyttöliittymän kautta.

Käyttöliittymä on selkeä ja looginen. Se ei muistuta mitään toista käyttöliittymää, joka testeissäni on. Proxmox on suoraan kilpailija VMwarelle, sillä se ei keskity pilvessä virtualisoimiseen. Mikäli ylläpitäjä haluaa paljon tietoa käyttöliittymästä, eikä pilvi kiinnosta, niin silloin Proxmoxiin kannattaa tutustua.

8.3.5 oVirt

oVirt on eniten Proxmoxin kaltainen käyttöliittymältään. Asettelu on ylläpitäjälle Proxmoxin kaltainen. Samoista paikoista löytyy samoja valikoita, mutta ero syntyy siitä, että oVirtissä on useita portaleja. Käyttäjälle se ei välttämättä tarjoa enempää näkymiä kuin Eucalyptuksessa. Toinen näkymä puolestaan tarjoaa esimerkiksi templatet näkyviin. Virtuaalikoneita voi käyttää suoraan selaimesta toisin kuin Eucalyptuksessa. Tämä koskee myös työpöytävirtualisointia.

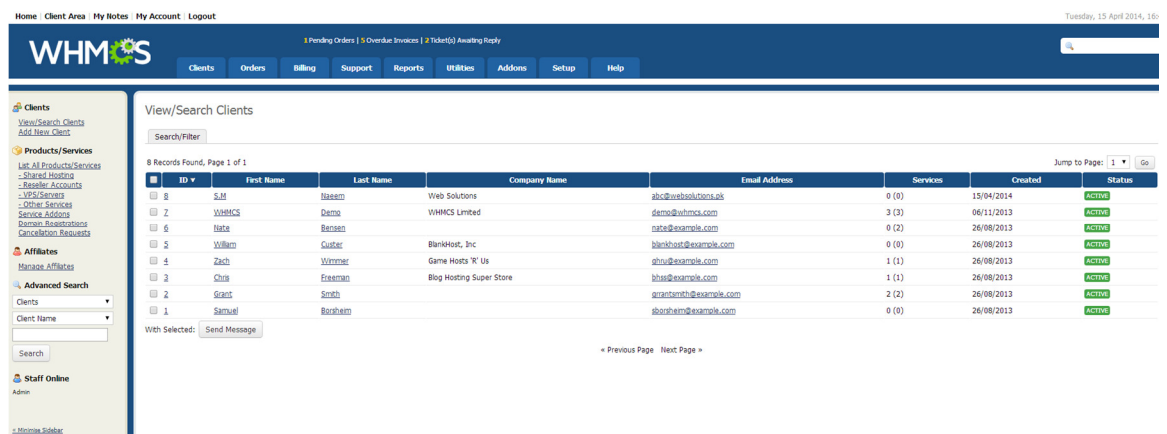
Ylläpitäjän ei ole pakko käyttää komentoriviä konfigurointia enempää, vaan hostien ylläpito onnistuu käyttöliittymästä. Logit eivät aina anna tarpeeksi tietoa, vaan front-endin logikansiota täytyy selata itse. Logit sisältävät paljon tekstiä, mikä tekee ongelman paikantamisesta hankalaa.

oVirt on pätevä käyttöliittymältään. Se ei hukkaa tilaa turhaan ja valikot löytyvät helposti niin käyttäjälle kuin ylläpitäjällekin. oVirt sopii käytettäväksi tilanteessa, jossa halutaan käyttäjälle

yksinkertainen näkymä ja ylläpitäjälle helppo työkalu ympäristön tarkkailuun ja hallitsemiseen käyttöliittymän kautta hyvillä ominaisuuksilla varustettuna.

8.4 Asiakkaiden hallinta

Virtualisointiympäristön tai virtuaalikoneiden vuokraaminen on yksi merkittävä asia joillekin ohjelmistoille ja ylläpitäjille. OpenNode on integroitunut WHMCS-ohjelmistoon, kuten myös Proxmox. oVirt ei tue minkäänlaista maksu/asiakkuudenhallinta ohjelmistoa. OpenNebulan kohdalla Accountin Client työkalusetti on olemassa OpenNebulaan, jolla integrointi vaikkapa WHMCS:n kanssa on teoriassa mahdollista. Eucalyptus on tukenut versiosta 3.0 lähtien RAC-ominaisuuden, jota voidaan käyttää vastaaviin järjestelmiin. Kuvassa 36 esitetään WHMCS:n käyttöliittymää. (Eucalyptus g 2014), (OpenNebula h 2014), (WHMCS 2014)



The screenshot shows the WHMCS 'View/Search Clients' interface. It includes a sidebar with navigation links like 'Clients', 'Products/Services', and 'Affiliates'. The main area displays a table of 8 client records. Each record includes an ID, first and last name, company name, email address, number of services, creation date, and status (Active or Pending).

ID	First Name	Last Name	Company Name	Email Address	Services	Created	Status
8	Sam	Neem	Web Solutions	abc@webolutions.co	0 (0)	15/04/2014	Pending
7	WHMCS	Demo	WHMCS Limited	demo@whmcs.com	3 (3)	06/11/2013	Active
6	Nate	Benson		nate@example.com	0 (2)	26/08/2013	Active
5	William	Cutter	BlankHost, Inc	blankhost@example.com	0 (0)	26/08/2013	Active
4	Zach	Whinner	Game Hosts 'R' Us	zhu@example.com	1 (1)	26/08/2013	Active
3	Chris	Freeman	Blog Hosting Super Store	chris@example.com	1 (1)	26/08/2013	Active
2	Grant	Smith		grantsmith@example.com	2 (2)	26/08/2013	Active
1	Samuel	Borham		borham@example.com	0 (0)	26/08/2013	Active

Kuva 36. WHMCS:n demoversion käyttöliittymä.

Mikäli ylläpitäjän tarkoitus on vuokrata palvelimiaan ulkomailmaan, niin teknisesti on päästävää sille tasolle, että se on ylipäänsä mahdollista. Eucalyptus, oVirt, OpenNebula ja Proxmox ovat ainakin teoriassa vaihtoehtoja, joita kannattaa katsoa. Suuressa mittakaavassa kelkasta putoavat oVirt ja Proxmox, koska niiden tekniset ominaisuudet ovat heikommat. Tällä tarkoitan load-balancing- ja auto-scaling-ominaisuuksien puutetta. Käytännössä vaihtoehtoiksi jäävät Eucalyptus ja OpenNebula.

8.5 KVM vastaan VMware

Mikäli KVM-hallintatyökaluja verrataan VMwaren tuotteisiin, niin välttämättä yhtä kattavaa pakettia ei mistään testissani olleista työkaluista saa. VMwaren vahvuudet ovat siinä, että lähes kaikessa siltä löytyy ainakin yksi ominaisuus enemmän ja usein paremmin tai helpommin toteutettuna ylläpitäjän ja käyttäjän näkökulmasta. Se missä eroavaisuudet näkyvät eniten riippuu siitä, mihin hallintatyökaluun sitä verrataan.

8.5.1 VMwaren vahvuudet

Suurimpana vahvuutena VMwarella pidän virtuaalikoneeseen yhdistämisen ja niiden luomisen helppoutta. KVM-hallintatyökaluissa se on suurin ongelma. Varsinkin libvirt-rajapintaa käyttävät ohjelmistot perustuvat siihen, että ensin tehdään template ja sitten siitä luodaan virtuaalikone. Tämä prosessi ei välttämättä ole huono asia, mutta pakollinen vaihe se silti on. VMwarella ja Proxmoxilla virtuaalikone voidaan luoda suoraan asennusmediasta ilman templaten luomista niillä resursseilla, mitä milloinkin halutaan. Tällöin templateja ei tarvitse tehdä useita pienimpiinkin asioihin.

Ominaisuuksien valossa VMware on kärkipäässä, mutta OpenNebula ja Eucalyptus ovat kehittyneet perässä hyvin paljon. Esimerkiksi load-balancing ja auto-scaling ominaisuudet löytyvät kummastakin. Proxmox voisi mielestäni nousta haastajaksi ainakin käytettävyyden puolesta, jos siihen lisättäisiin lisää ominaisuuksia.

VMware tuottaa myös itse työkalunsa. Se ei ole samoissa määrin riippuvainen kolmannen osapuolen toiminnasta ominaisuuksia kehitettäessä. Avoimen lähdekoodin puolella kaikki ohjelmistot turvautuvat johonkin toiseen avoimen lähdekoodin kehitysprojektiin kuten esimerkiksi vSwitch tai libvirt.

8.5.2 VMwaren heikkoudet

VMware on ominaisuuksiltaan vahva, mutta sillä on hintansa. Siinä missä KVM-ratkaisut ovat teoriassa ilmaisia, niin VMwaresta joutuu maksamaan lisenssimaksuja. VMwaren oma

laskuri osoitteessa: "<http://www.VMware.com/go/tccalculator/>" kertoo, että virtualisoinnin lisenssikulut ovat suuri osa infrastruktuurin maksuista. Kokonaiskulut riippuvat hyvin paljon ympäristön koosta.

Toinen VMwaren heikkous on sen sulkeutuneisuus. Perustan tämän väitteen siihen, että VMwaren ympäristö tukee vain VMwaren tuotteita, joka pakottaa ylläpitäjän pitäytymään niissä. VMware on kuitenkin aloittanut tukemaan kolmannen osapuolen virtualisointiratkaisuja kehittämällä Multi-Hypervisor Managerin, jolla voidaan hallita muita hypervisoreita. Tu-ki kuitenkin rajoittuu toistaiseksi vain Hyper-V alustoihin. Tämäkään ei siis laske kuluja, koska Hyper-V ei ole ilmainen. (VMware 2014)

8.5.3 Avoimen lähdekoodin ohjelmistojen vahvuudet

Avoimen lähdekoodin ohjelmistot ovat ilmaisia. Osa ohjelmistojen tarjoajista tarjoaa maksullista tukea subscriptionin muodossa, mutta itse ohjelman käyttäminen ei maksa mitään. Tätä kautta ainakin virtualisoinnin lisenssimaksut vähenevät.

Avoimen lähdekoodin ohjelmistot tarjoavat vaihtoehtoja. Ohjelmistoja löytyy pilvipalveluksi saakka tavallisesta virtualisointi ympäristön hallinnasta. Esimerkiksi maksuhallintaohjelman liittämällä virtuaalipalvelimia voi vaikka vuokrata toisille, jos se on tarpeen. Myös teknologian virtualisoinnin taustalla voi valita itse käyttämällä esimerkiksi Proxmoxia tai eri tekniikalla toteutettua oVirtia.

Avoimen lähdekoodin ohjelmistoilla voi hallita myös muita hypervisoreita. Ympäristön liittäminen VMwaren ympäristön päälle onnistuu esimerkiksi OpenNebulalla, joka tukee ESXi-hypervisoreita. Tällöin hallintatyökalu vaihtuu, mutta virtuaalikoneet ovat edelleen samat. ESXi-palvelimia voi tällöin käyttää myös KVM-palvelimien kanssa samassa hallintatyökalussa.

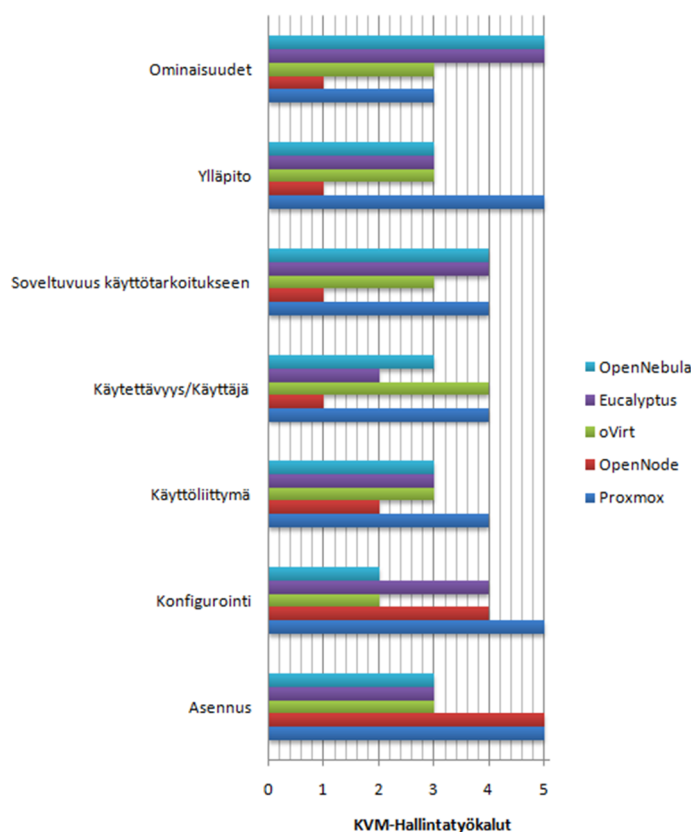
8.5.4 Avoimen lähdekoodin ohjelmistojen heikkoudet

Avoimen lähdekoodin hallintaohjelmistojen kohdalla kysymykseksi nousee tuki. Mikäli ongelmatilanne ilmenee, niin siihen saattaa tarvita apua esimerkiksi kehittäjältä. Näissä tapauksissa ylläpitäjä joutuu usein luottamaan mail-listoihin, sähköpostiin ja IRC-kanavaan. Osalla ammattimaisemmin asiansa hoitavilla kehittäjillä on kuitenkin oma tuki kysymyksille joko puhelimitse, etänä tai paikanpäälle tilaamana.

Toinen heikkous on kehitystyö. Avoimen lähdekoodin työkalut ovat riippuvaisia toisen porukan työskentelystä toisessa projektissa. Esimerkiksi jos oVirt haluaisi tuoda jonkin ominaisuuden ohjelmistoonsa, niin se ei välttämättä onnistu jos sitä ei ole tuettuna libvirt-rajapinnassa. Tällöin heidän on odotettava sitä, että libvirt lisää kyseisen ominaisuuden ohjelmaansa.

8.6 Testaajan tulokset

Testattavat ohjelmistot tarjoavat erilaisia ratkaisuja erilaisiin ympäristöihin samankaltaisilla keinoilla. Selvää yhdenmukaisuutta ohjelmistojen suunnittelussa selvästi on, sillä kaikista niistä löytyy ratkaisuja jotka ovat identtisiä jonkun toisen ohjelmiston kanssa. Kuvassa 37 esitetään testitulokset.



Kuva 37. KVM-hallintatyökalut arvostelussa

Testeissäni parhaiten suoriutui Proxmox. Proxmoxia on helppo ylläpitää ja käyttää. Proxmoxin pystyttäminen oli niin nopeaa, että ehdin asennuksessa tutustua HA-mahdollisuuksiin onnistuneesti. Proxmoxin kohdalla on myös hyvä huomioida, että se eroaa muista hallintatyökaluista omalla API-tekniikalla, jolloin se ei ole riippuvainen libvirt-kehityksestä. Lisäksi Proxmoxissa on potentiaalia muita hallintatyökaluja enemmän, koska se on tällä hetkellä todella toimiva paketti. Proxmoxin tulevaisuus näyttää hyvältä, jos siihen lisätään suuren datacenterin vaatimia ominaisuuksia. Proxmox on pieneen ympäristöön sopiva virtualisointityökalu ainakin tämän hetken ominaisuuksiltaan.

Toiseksi testeissäni suoriutui OpenNebula. Sen kohdalla asennustyö oli huomattavasti vaikeampi kuin Proxmoxissa. Ominaisuuksiltaan OpenNebula oli testien paras, sillä se pitää sisällään suurempiin ympäristöihin soveltuvia load-balancing ja auto-scaling ominaisuuksia, joita Proxmoxissa ei ole. OpenNebula on pilvipalvelu, joten sen käyttöä pelkän virtualisointialustan hallintaan kannattaa miettiä. OpenNebula panostaa virtualisointialustan hallintaan paljon, mutta sen käyttöliittymä ei välttämättä ole siihen paras mahdollinen. Valitsisin OpenNebulan KVM-hallintatyökaluksi, jos hallittava ympäristö olisi suuri.

Kolmanneksi paras testaamani työkalu on Eucalyptus. Eucalyptuksen ominaisuudet vastaavat OpenNebulan ominaisuuksia, mutta eroa syntyy siitä, että Eucalyptus vie pilveä pidemmälle. Se ei missään tapauksessa ole paras vaihtoehto virtualisointialustan hallintatyökaluksi käyttöliittymän yksinkertaisuuden vuoksi, mutta jos ympäristöstä halutaan pilveen perustuva, niin se olisi silloin ykkösvalinta varsinkin, jos AWS:n ominaisuuksia kaivataan. Käyttöliittymän yksinkertaisuus heijastuu myös ylläpitoon, sillä ylläpitäjän täytyy käyttää kolmannen osapuolen monitorointityökaluja hostien monitoroimiseen. Mikäli datacenterin hallinnan tärkeys ei ole pääasia, vaan halutaan pilven ominaisuuksia enemmän, niin Eucalyptus on hyvä vaihtoehto AWS-käyttöön.

Neljänneksi sijoittui oVirt. oVirt muistuttaa Proxmoxia, mutta käyttää libvirtä API:na. Sen konfiguroimisessa oli suuria vaikeuksia ja vian havaitseminen vaikeaa, koska logit olivat täynnä virheitä. Login selaamisessa huomasin, että kun yksi asia ei toimi, niin kaikki muutkin asiat työnsivät niin paljon herjaa, että ”harmituksen aste” oli suuri. Täydessä iskussa oleva oVirt on vaihtoehto KVM-hallintatyökaluksi datacenter ympäristöön, mutta se ei ole yhtä kattava ominaisuuksiltaan kuin OpenNebula tai Eucalyptus, jotta laittaisin sitä suuren ympäristöön. oVirt soveltuu pienen ympäristön hallintatyökaluksi, mutta se ei haasta ainakaan nykyisessä tilassaan Proxmoxia. Proxmoxin käyttöliittymä on hiotumpi ja ympäristö helpommin ylläpidettävä. oVirtin ongelma on lähinnä pienien asioiden konfigurointi toimintakuntoon. Versioiden kanssa tulee myös ongelmia, joka ilmenee siinä, että nodeja ei saa helposti hostiin kiinni. Ominaisuudet ovat lähestulkoon samat, mutta Proxmox tuntuu vakaammalta ja käyttöliittymältään datacenterin suuntaiselta. oVirtin kehityskaari on kilpailijoita lyhempi, joten siitä voi tulla vielä laadukas ohjelmisto, mikäli kehityksen suunta pysyy samana.

Viimeiseksi testeissäni sijoittui OpenNode. Viimeinen sija on ansaittu, sillä se ei tarjoa KVM-hallintaan mitään. KVM-tuki oli olemassa vain siltä osin, että ympäristössä olisi ollut valmis KVM-virtuaalikone, josta voi tehdä templatien. Ominaisuudet jotka olivat tarjolla OpenVZ:lle, eivät olleet tarjolla KVM:lle ainakaan toistaiseksi. OpenNodea en voi suositella muuhun kuin harrastekäyttöön.

LÄHTEET

Cloudbook.net. a. Cloud Controller. 2014. Saatavilla:
<http://www.cloudbook.net/resources/stories/the-eucalyptus-open-source-private-cloud>
 (Luettu 17.4.2014)

Cloudbook.net. b. Cluster Controller. 2014. Saatavilla:
<http://www.cloudbook.net/resources/stories/the-eucalyptus-open-source-private-cloud>
 (Luettu 17.4.2014)

Eucalyptus. a. The Eucalyptus Story. 2014 Saatavilla:
<https://www.eucalyptus.com/about/story> (Luettu 15.4.2014)

Eucalyptus. b. Get Started With A Eucalyptus Private Cloud. 2014 Saatavilla:
<https://www.eucalyptus.com/eucalyptus-cloud/get-started/try> (Luettu 15.4.2014)

Eucalyptus. c. Configuring the SC to use the local filesystem (Overlay). 2014 Saatavilla:
https://www.eucalyptus.com/docs/eucalyptus/3.2/ig/configure_storage_controller.html#enabling_overlay (Luettu 15.4.2014)

Eucalyptus. d. VMware Support. 2014. Saatavilla:
https://www.eucalyptus.com/docs/eucalyptus/3.2/ig/planning_vmware.html#planning_vmware (Luettu 17.4.2014)

Eucalyptus. e. Using CloudWatch. 2014 Saatavilla:
https://www.eucalyptus.com/docs/eucalyptus/3.3/user-guide/using_monitoring.html#using_monitoring (Luettu 15.4.2014)

Eucalyptus f. Eucalyptus Cloud 4.0 Roadmap And Timeline. 2014 Saatavilla:
<https://www.eucalyptus.com/eucalyptus-cloud/iaas/roadmap/eucalyptus-4-0> (Luettu 17.4.2014)

Eucalyptus. g. Eucalyptus Launches The Third Generation Of Its Enterprise Private Cloud Software. 2014 Saatavilla: <https://www.eucalyptus.com/news/eucalyptus-launches-third-generation-its-enterprise-private-cloud-software> (Luettu 15.4.2014)

Keningsberg, D. Vdsm. 2011. Saatavilla: <http://www.linux-kvm.org/wiki/images/6/66/Vdsm.pp.pdf> (Luettu 17.4.2014)

Lesovsky, A. Getting Started with oVirt 3.3. 2013. Birmingham: Packt Publishing Ltd.

libvirt. a. What is libvirt. 2014. Saatavilla: http://wiki.libvirt.org/page/FAQ#What_is_libvirt.3F (Luettu 15.4.2014)

libvirt b. libvirt is. 2014 Saatavilla: <http://libvirt.org/index.html> (Luettu 15.4.2014)

Lists.opennebula.org. libvirt and kvm permission problem. 2011. Saatavilla: <http://lists.opennebula.org/pipermail/users-opennebula.org/2011-March/014614.html> (Luettu: 23.4.2014)

Mail-archive.com. [Users] ComputeNode installation failed ovirt 3.3. 2013. Saatavilla: <https://www.mail-archive.com/users@ovirt.org/msg10839.html> (Luettu 15.4.2014)

OpenNebula a. About the OpenNebula project. 2014. Saatavilla: <http://opennebula.org/about/project/> (Luettu 15.4.2014)

OpenNebula. b. KVM Sandbox. 2013. Saatavilla: <http://opennebula.org/tryout/sandboxkvm/> (Luettu:13.11.2013)

OpenNebula c. OneFlow. 2014. Saatavilla: http://archives.opennebula.org/documentation:archives:rel4.2:oneapps_overview (Luettu 15.4.2014)

OpenNebula. d. OpenNebula Zones Overview 3.0. 2014. Saatavilla: <http://archives.opennebula.org/documentation:archives:rel3.0:ozones> (Luettu:23.4.2014)

OpenNebula. e. What is the OpenNebula Technology 2014. Saatavilla: <http://opennebula.org/about/technology/> (Luettu 15.4.2014)

OpenNebula. f. What are our Design Principles?. 2014. Saatavilla: <http://opennebula.org/about/technology/> (Luettu 15.4.2014)

OpenNebula. g. How Does OpenNebula compare with other Cloud platforms?. 2014. Saatavilla: <http://opennebula.org/about/faq/#toggle-id-2> (Luettu 15.4.2014)

- OpenNebula. h. Accounting Client 4.4. 2014. Saatavilla: <http://archives.opennebula.org/documentation:rel4.4:accounting> (Luettu 15.4.2014)
- OpenNode. a. OpenNode Cloud Platform. 2014. Saatavilla: <http://opennodecloud.com/about/> (Luettu 15.4.2014)
- OpenNode. b. Partners and integrations. 2014. Saatavilla: <http://opennodecloud.com/about/partners> (Luettu 15.4.2014)
- OpenNode. c. OpenNode Cloud Platform advantages. 2014. Saatavilla: <http://opennodecloud.com/about/why-opennode/> (Luettu 15.4.2014)
- OpenNode. d. Architecture overview. 2014. Saatavilla: <http://opennodecloud.com/about/architecture/> (Luettu: 23.4.2014)
- OpenNode. e. Prerequisites. 2014. Saatavilla: <https://support.opennodecloud.com/wiki/doku.php?id=usrdoc:qsguide> (Luettu 15.4.2014)
- OpenNode. f. Roadmap. 2014. Saatavilla: <http://opennodecloud.com/about/roadmap/> (Luettu 15.4.2014)
- OpenNode. g. WHMCS billing software integration 2014. Saatavilla: <http://opennodecloud.com/about/whmcs-integration/> (Luettu 15.4.2014)
- oVirt. Projects. 2014. Saatavilla: <http://www.ovirt.org/Develop> (Luettu 15.4.2014)
- Proxmox. a. Configuring Fencing. 2014. Saatavilla: http://pve.proxmox.com/wiki/Two-Node_High_Availability_Cluster (Luettu 15.4.2014)
- Proxmox. b. About Proxmox. 2014. Saatavilla: <https://www.proxmox.com/about> (Luettu 15.4.2014)
- Proxmox. c. Proxmox VE Cluster. 2014. Saatavilla: http://pve.proxmox.com/wiki/Proxmox_VE_2.0_Cluster#Proxmox_VE_Cluster (Luettu 15.4.2014)
- Proxmox. d. User Management. 2014. Saatavilla: http://pve.proxmox.com/wiki/User_Management (Luettu 15.4.2014)

SPICE. Etusivu. 2014. Saatavilla: <http://www.spice-space.org/> (Luettu 15.4.2014)

Tolaris.com. Removing the Proxmox 3.1 subscription nag dialog. 2013. Saatavilla: <http://www.tolaris.com/2013/08/28/removing-the-proxmox-3-1-subscription-nag-dialog/> (Luettu 14.11.2013)

VMware. What is VMware vCenter Multi-Hypervisor Manager 1.1. 2014. Saatavilla: <https://www.VMware.com/support/mhm/doc/vcenter-multi-hypervisor-manager-11-release-notes.html#whatis> (Luettu 15.4.2014).

WHMCS. Proxmox Full Automation. 2014. Saatavilla <https://www.whmcs.com/appstore/517/Proxmox-Full-Automation.html> (Luettu 15.4.2014)